

JC10 Rec'd PCT/PTO 07 MAR 2002

FORM PTO-1390 REV. 5-93		US DEPARTMENT OF COMMERCE PATENT AND TRADEMARK OFFICE	ATTORNEYS DOCKET NUMBER P02,0086
TRANSMITTAL LETTER TO THE UNITED STATES DESIGNATED/ELECTED OFFICE (DO/EO/US) CONCERNING A FILING UNDER 35 U.S.C. 371			U.S. APPLICATION NO. (if known, see 37 CFR 1.5) 10/070701
INTERNATIONAL APPLICATION NO. PCT/SE00/01744	INTERNATIONAL FILING DATE September 7, 2000	PRIORITY DATE CLAIMED September 8, 1999	
TITLE OF INVENTION: "COMPRESSION AND DECOMPRESSION CODING SCHEME AND APPARATUS"			
APPLICANT(S) FOR DO/EO/US: JOHAN LIDMAN			
Applicant herewith submits to the United States Designated/Elected Office (DO/EO/US) the following items and other information:			
<ol style="list-style-type: none"> 1. <input checked="" type="checkbox"/> This is a FIRST submission of items concerning a filing under 35 U.S.C. 371. 2. <input type="checkbox"/> This is a SECOND or SUBSEQUENT submission of items concerning a filing under 35 U.S.C. 371. 3. <input checked="" type="checkbox"/> This express request to begin national examination procedures (35 U.S.C. 371(f)) at any time rather than delay. 4. <input checked="" type="checkbox"/> A proper Demand for International Preliminary Examination was made by the 19th month from the earliest claimed priority date. 5. <input checked="" type="checkbox"/> A copy of International Application as filed (35 U.S.C. 371(c)(2)) <ol style="list-style-type: none"> a. <input checked="" type="checkbox"/> is transmitted herewith (required only if not transmitted by the International Bureau). b. <input type="checkbox"/> has been transmitted by the International Bureau. c. <input type="checkbox"/> is not required, as the application was filed in the United States Receiving Office (RO/US) 6. <input checked="" type="checkbox"/> A translation of the International Application into English (35 U.S.C. 371(c)(2)). 7. <input checked="" type="checkbox"/> Amendments to the claims of the International Application under PCT Article 19 (35 U.S.C. §371(c)(3)) <ol style="list-style-type: none"> a. <input type="checkbox"/> are transmitted herewith (required only if not transmitted by the International Bureau). b. <input type="checkbox"/> have been transmitted by the International Bureau. c. <input checked="" type="checkbox"/> have not been made; however, the time limit for making such amendments has NOT expired. d. <input type="checkbox"/> have not been made and will not be made. 8. <input type="checkbox"/> A translation of the amendments to the claims under PCT Article 19 (35 U.S.C. 371(c)(3)). 9. <input checked="" type="checkbox"/> An oath or declaration of the inventor(s) (35 U.S.C. 371(c)(4)). (UNSIGNED) 10. <input type="checkbox"/> A translation of the annexes to the International Preliminary Examination Report under PCT Article 36 (35 U.S.C. 371(c)(5)). 			
Items 11. to 16. below concern other document(s) or information included:			
<ol style="list-style-type: none"> 11. <input checked="" type="checkbox"/> An Information Disclosure Statement under 37 C.F.R. 1.97 and 1.98; (PTO 1449, Prior Art, Search Report). 12. <input type="checkbox"/> An assignment document for recording. A separate cover sheet in compliance with 37 C.F.R. 3.28 and 3.31 is included. 13. <input checked="" type="checkbox"/> A FIRST preliminary amendment. <ol style="list-style-type: none"> <input type="checkbox"/> A SECOND or SUBSEQUENT preliminary amendment. 14. <input checked="" type="checkbox"/> A substitute specification. 15. <input type="checkbox"/> A change of power of attorney and/or address letter. 16. <input checked="" type="checkbox"/> Other items or information: <ol style="list-style-type: none"> a. <input checked="" type="checkbox"/> Submission of Informal Drawings; Request For Approval of Drawing Changes and Priority Document 9903191-6 b. <input checked="" type="checkbox"/> EXPRESS MAIL # EJ 552525055US 			

U.S. APPLICATION NO (if known, see 37 C.F.R. 1.5)		INTERNATIONAL APPLICATION NO.		ATTORNEY'S DOCKET NUMBER	
10/070701		PCT/SE00/01744		P02,0086	

<p>17. ■ The following fees are submitted:</p> <p>BASIC NATIONAL FEE (37 C.F.R. 1.492(a)(1)-(5): Neither international preliminary examination fee (37 C.F.R. 1.482) nor international search fee (37 C.F.R. 1.445(a)(2)) paid to USPTO and International Search Report not prepared by EPO or JPO \$1040.00</p> <p>No international preliminary examination fee USPTO (37 C.F.R. 1.482) not paid to USPTO but International Search Report prepared by the EPO or JPO \$890.00</p> <p>International preliminary examination fee USPTO (37 C.F.R. 1.482) not paid to USPTO but international search fee fee paid to USPTO (37 C.F.R. 1.445(a)(2)) \$740.00</p> <p>International preliminary examination fee paid to USPTO (37 C.F.R. 1.482) but all claims did not satisfy provisions of PCT Article 33(1)-(4) \$710.00</p> <p>International preliminary examination fee paid to USPTO (37 C.F.R. 1.482) and all claims satisfied provisions of PCT Article 33(2)-(4) \$100.00</p> <p style="text-align: right;">ENTER APPROPRIATE BASIC FEE AMOUNT =</p>	CALCULATIONS	PTO USE ONLY

Surcharge of \$130.00 for furnishing the oath or declaration later than <input type="checkbox"/> 20 <input type="checkbox"/> 30 months from the earliest claimed priority date (37 C.F.R. 1.492(e))				\$	
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--	--	--	----	--

Claims	Number Filed	Number Extra	Rate		
Total Claims	17 - 20 =		X \$ 18.00	\$	
Independent Claims	3 - 3 =		X \$ 84.00	\$	
Multiple Dependent Claims			\$280.00 +	\$	
TOTAL OF ABOVE CALCULATIONS =				\$ 890.00	
Reduction by 1/2 for filing by small entity, if applicable. Verified Small Entity statement must also be filed. (Note 37 C.F.R. 1.9, 1.27, 1.28)				\$	
SUBTOTAL =				\$ 890.00	
Processing fee of \$130.00 for furnishing the English translation later than <input type="checkbox"/> 20 <input type="checkbox"/> 30 months from the earliest claimed priority date (37 CFR 1.492(f))				\$	
TOTAL NATIONAL FEE =				\$ 890.00	
Fee for recording the enclosed assignment (37 C.F.R. 1.21(h)). The assignment must be accompanied by an appropriate cover sheet (37 C.F.R. 3.28, 3.31). \$40.00 per property (see separate envelope)				\$	
TOTAL FEES ENCLOSED =				\$ 890.00	
				Amount to be refunded	\$
				charged	\$

a. ☐ A check in the amount of \$ _ to cover the above fees is enclosed.

b. ☐ Please charge my Deposit Account No. _____ in the amount of \$ _____ to cover the above fees. A duplicate copy of this sheet is enclosed.

c. ☒ The Commissioner is hereby authorized to charge any additional fees which may be required, or credit any overpayment to Deposit Account No. **501519**. A duplicate copy of this sheet is enclosed.

d. ☐ Fees are to be charged to a credit card. **WARNING:** Information on this form may become public. **Credit card information should not be included on this form.** Provide credit card information and authorization on PTO-2038

NOTE: Where an appropriate time limit under 37 C.F.R. 1.494 or 1.495 has not been met, a petition to revive (37 C.F.R. 1.137(a) or (b)) must be filed and granted to restore the application to pending status.

SEND ALL CORRESPONDENCE TO:

Schiff, Hardin & Waite
CUSTOMER NO. 26574
Patent Department
6600 Sears Tower
233 South Wacker Drive
Chicago, Illinois 60606

SIGNATURE

NAME

Registration Number

BOX PCT
IN THE UNITED STATES DESIGNATED OFFICE
OF THE UNITED STATES PATENT AND TRADEMARK OFFICE
UNDER THE PATENT COOPERATION TREATY-CHAPTER II
5 AMENDMENT "A" PRIOR TO ACTION AND SUBMISSION OF
SUBSTITUTE SPECIFICATION

APPLICANT: Johan Lidman
ATTORNEY DOCKET NO. P02,0086
INTERNATIONAL APPLICATION NO: PCT/SE00/01744
10 INTERNATIONAL FILING DATE: September 7, 2000
INVENTION: "COMPRESSION AND DECOMPRESSION CODING
SCHEME AND APPARATUS"

Assistant Commissioner for Patents
Washington, D.C. 20231

15 Sir:

Applicant herewith amends the above-referenced PCT application as follows, and requests entry of the Amendment prior to examination in the United States National Examination Phase.

IN THE SPECIFICATION:

20 Please enter the substitute specification submitted herewith pursuant to 37 C.F.R. §1.125(b). A marked-up copy of the substitute specification, showing all of the changes, is also submitted herewith. The substitute specification does not add any new matter.

IN THE DRAWINGS:

25 Please amend each of Figures 1, 2 and 8, and make the further changes on each of drawing sheets 1 through 6, as shown on the drawing copies marked in red attached to the Request for Approval of Drawing Changes filed simultaneously herewith.

IN THE CLAIMS:

30 On page 16, in line 1, cancel "Claims" and substitute:
--I CLAIM AS MY INVENTION:-- therefor.

-2-

Cancel claims 1-17 and substitute the following claims therefor:

18. A data coding method comprising the steps of:

monitoring a data signal containing a plurality of symbols and
determining a plurality of most frequently occurring data
components in said data signal, selected from the group
consisting of most frequently occurring symbols and most
frequently occurring sequences of symbols containing at least
two symbols;

allocating respective codewords to said most frequently occurring
data components, thereby obtaining a codeword set; and
forming a compressed signal by substituting the respective
codewords for said most frequently occurring data
components.

19. A method as claimed in claim 18 wherein the step of monitoring
said data signal comprises monitoring said data signal during a
predetermined time period.

20. A method as claimed in claim 18 wherein said data signal
includes uncoded symbols, that are not among said plurality of most
frequently occurring symbols, and comprising the additional step of reserving
at least one codeword in said set as an indicator for said uncoded symbols.

21. A method as claimed in claim 20 wherein said uncoded
symbols include uncoded negative symbols, and comprising supplementing
said at least one codeword serving as said indicator for uncoded symbols
with at least one further codeword, for said uncoded negative symbols,
indicative of a negative value.

22. A method as claimed in claim 18 wherein the step of allocating
codewords comprises allocating codewords to respective data components
that are incorporated in other data components having another codeword
allocated thereto.

-3-

23. A data compression method comprising the steps of:
 converting a plurality of most frequently occurring data components
 in a data signal containing a plurality of symbols into respective
 codewords, said most frequently occurring data components
 5 being selected from the group consisting of most frequently
 occurring symbols and most frequently occurring sequences of
 symbols containing at least two symbols; and
 designating remaining symbols in said data signal, not among said
 most frequently occurring data components, with at least one
 10 codeword indicative of no compression; and
 substituting said codewords in place of said symbols.

24. A method as claimed in claim 23 comprising setting a
 predetermined number and a predetermined length for said codewords.

25. A method as claimed in claim 23 comprising preprocessing an
 15 input signal containing a plurality of symbols to generate said data signal by
 generating an additional symbol representing a difference between
 contiguous symbols in said input signal.

26. A method as claimed in claim 23 comprising the additional
 steps of:
 20 reading a symbol in said data signal;
 determining if the symbol that has been read corresponds to a
 codeword; and
 substituting said codeword for said symbol that has been read if said
 symbol that has been read corresponds to only one codeword.

27. A method as claimed in claim 26 wherein said symbol that has
 25 been read is a first symbol, and comprising the additional steps, if said first
 symbol corresponds to more than one codeword, of:

reading a subsequent symbol following said first symbol;
 determining if said first symbol and said subsequent symbol
 30 correspond to a codeword; and

-4-

substituting a codeword in place of said first symbol and said subsequent symbol if said first symbol and said subsequent symbol correspond to only one codeword.

28. A method as claimed in claim 27 comprising the additional
5 step, if said symbol that has been read corresponds to no codeword, retaining said symbol that has been read in said data signal.

29. An arrangement for compressing and decompressing a data signal, comprising:

10 a memory for storing codewords respectively corresponding to data components selected from the group consisting of symbols and symbol sequences; and

15 a determination unit supplied with a data signal containing a plurality of symbols for determining if a symbol in said data signal corresponds to a codeword in said memory and, if a symbol corresponds to only one codeword in said memory, transmitting that codeword in place of said symbol and transmitting said symbol if said symbol corresponds to no codeword in said memory.

20 30. An arrangement as claimed in claim 29 wherein said memory includes a plurality of memory locations respectively designating codewords, and wherein each memory location contains an indication of a number of possible symbol sequences, and is mapped to a symbol of said data signal.

25 31. An arrangement as claimed in claim 30 further comprising a difference symbol generator, connected preceding said determination unit, which generates a difference symbol between contiguous symbols in said data signal.

32. An arrangement as claimed in claim 29 wherein said memory comprises a plurality of memory locations having respective addresses, and wherein said addresses are said codewords.

-5-

33. A computer program product for converting a data signal containing a plurality of symbols into a compressed signal, said computer program product comprising:

5 a computer-readable program code for establishing a set of
codewords by determining a plurality of most frequently
occurring data components in a data signal, said most
frequently occurring data components being selected from the
group consisting of most frequently occurring symbols and
most frequently occurring sequences of symbols containing at
10 least two symbols; and
said program code allocating one codeword to each of said most
frequently occurring data components.

34. A computer program product as claimed in claim 33 wherein
said program code compresses said data signal by converting said most
15 frequently occurring data components into respective codewords by reading
a symbol in said data signal and determining if said symbol corresponds to
a codeword, and if so, emitting said codeword instead of said symbol and,
if not, emitting said symbol.

IN THE ABSTRACT:

20 Please add the Abstract shown separately numbered page 20,
attached hereto.

REMARKS:

25 The present Amendment makes editorial changes in the specification,
drawings, claims, and adds an Abstract to conform the present PCT
application to the requirements of United States patent practice. The
cancellation of claims 1-17 in favor of the claims presented herein has been
done solely because the amount of bracketing and underlining in the original
claims which would have been necessary to conform those claims to the

-6-

requirements of 35 U.S.C. §112, second paragraph, would have been unduly
burdensome and confusing. No change in any of the claim language has
been made for distinguishing any of the original claims over the teachings of
the art of record, accordingly no change in the language of any claim is
5 considered by the Applicant as a surrender of any coverage encompassed
within the scope of the original claims.

Early consideration of the PCT application is respectfully requested.

Submitted by,

Stan H. Noll

(Reg. 28,982)

10

SCHIFF, HARDIN & WAITE

CUSTOMER NO. 26574

Patent Department

6600 Sears Tower

233 South Wacker Drive

15

Chicago, Illinois 60606

Telephone: 312/258-5790

Attorneys for Applicant.

ABSTRACT OF THE DISCLOSURE

In a compression and decompression coding method, arrangement and computer program product, a data signal containing a number of symbols is converted into a series of codewords. A set of codewords is established and the data signal is monitored to determine the most frequently occurring symbols therein and/or sequences of symbols therein containing at least two symbols. A codeword is then allocated to each of the most frequently occurring of the symbols and/or symbol sequences. At least one codeword is reserved for indicating uncompressed data. When compressing a signal, the incoming symbols are first checked to determine if they correspond to a codeword. If a symbol corresponds to more than one codeword, further symbols are read until a symbol occurs which corresponds to one codeword only. That codeword is then transmitted. Any symbol that does not correspond to a codeword is supplemented with a codeword indicative of no compression and is then transmitted.

-1-

BOX PCT
IN THE UNITED STATES DESIGNATED OFFICE
OF THE UNITED STATES PATENT AND TRADEMARK OFFICE
UNDER THE PATENT COOPERATION TREATY-CHAPTER II

5 SUBMISSION OF DRAWINGS FOR PUBLICATION

APPLICANT: Johan Lidman

ATTORNEY DOCKET NO. P02,0086

INTERNATIONAL APPLICATION NO: PCT/SE00/01744

INTERNATIONAL FILING DATE: September 7, 2000

10 INVENTION: "COMPRESSION AND DECOMPRESSION CODING
SCHEME AND APPARATUS"

Assistant Commissioner for Patents,
Washington, D.C. 20231

S I R:

15 Applicant herewith submits six sheets (Figs. 1-11e) of drawings for publication purposes. The drawings embody the changes made in the Request for Approval of Drawing Changes, filed simultaneously herewith.

Submitted by,

Stan H. Noll (Reg. 28,982)

20 SCHIFF, HARDIN & WAITE
CUSTOMER NO. 26574

25 Patent Department
6600 Sears Tower
233 South Wacker Drive
Chicago, Illinois 60606
Telephone: 312/258-5790
Attorneys for Applicant.

Fig. 1
(PRIOR ART)

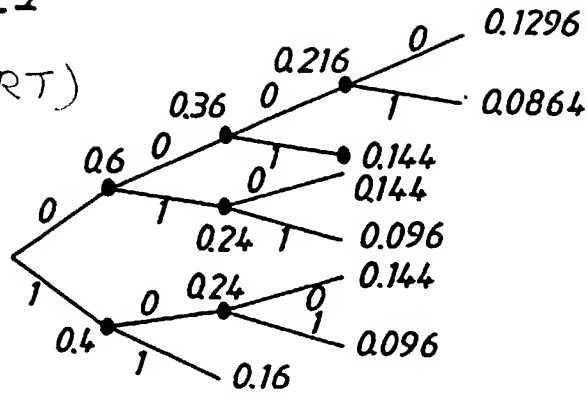


Fig. 2
(PRIOR ART)

Symbol sequences	Codeword	Codeword binary form
0000	1	001
0001	2	010
001	3	011
010	4	100
011	5	101
100	6	110
101	7	111
11	8	000

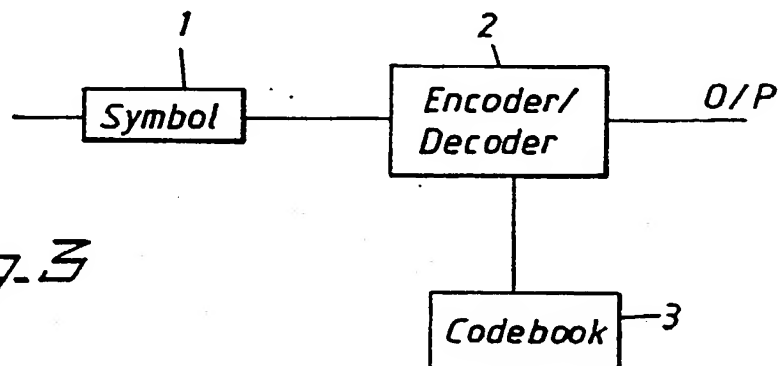


Fig. 3

2/6

Fig. 4a

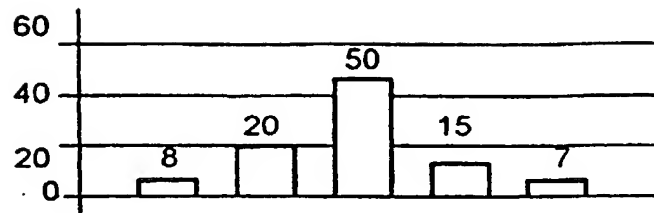


Fig. 4b

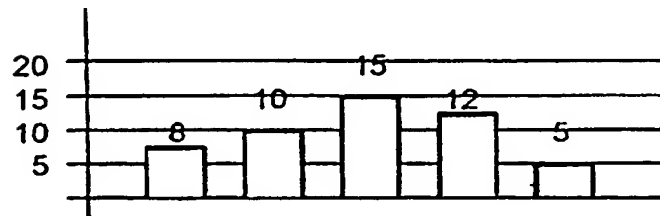


Fig. 4c

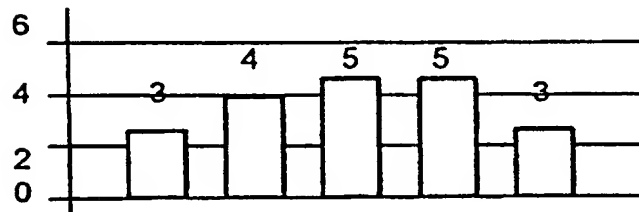


Fig. 4d

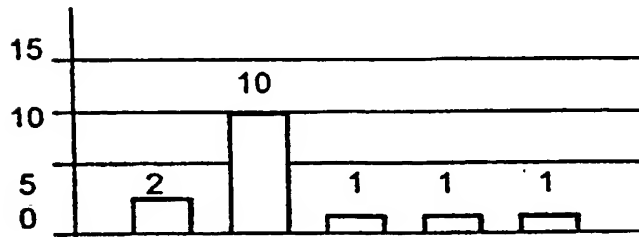


Fig. 4e

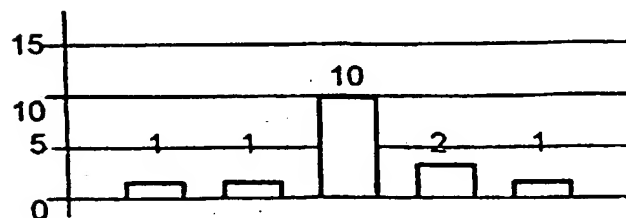


Fig. 5

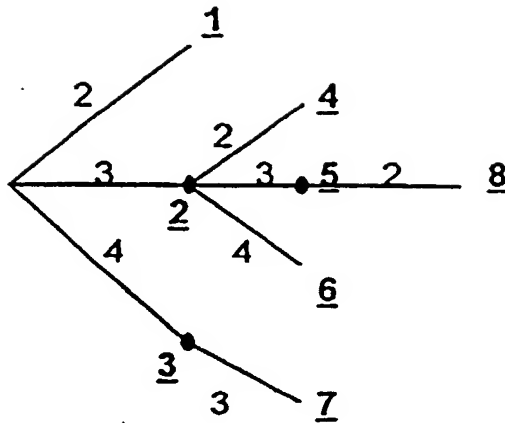
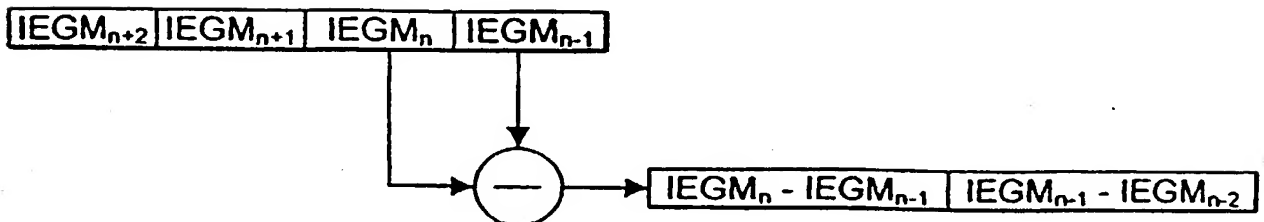


Fig. 6

Symbol sequence	Codeword
2	1
3	2
4	3
32	4
33	5
34	6
43	7
332	8

Fig. 7



4 / 6

Fig. 8

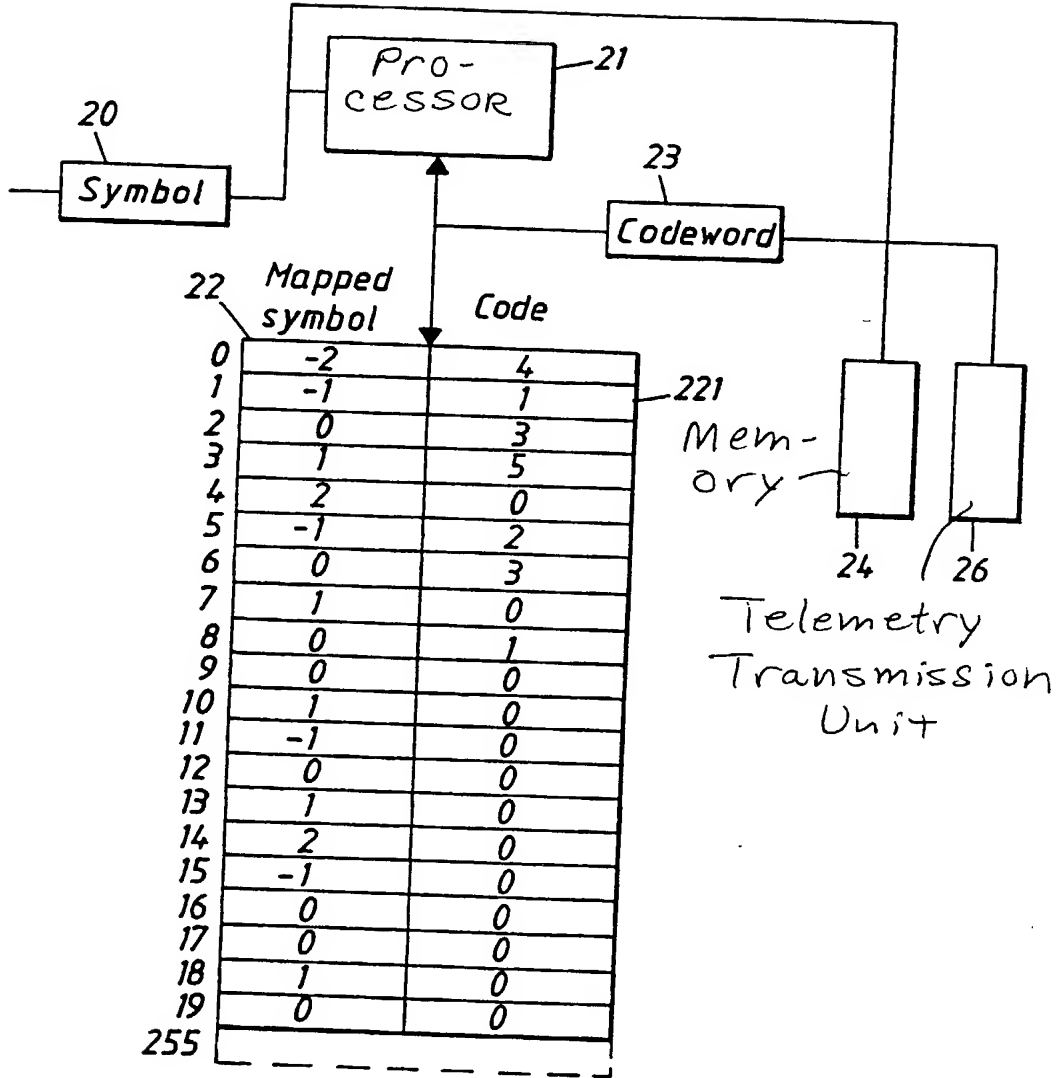


Fig. 9

25

Code No.	Possible branches	No. of branches
0	End	0
1	0	1
2	-1, 0	2
3	0, 1	2
4	-1, 0, 1	3
5	-1, 0, 1, 2	4
6	-2, -1, 0, 1	4
7	-2, -1, 0, 1, 2	5

5 / 6

Fig. 10

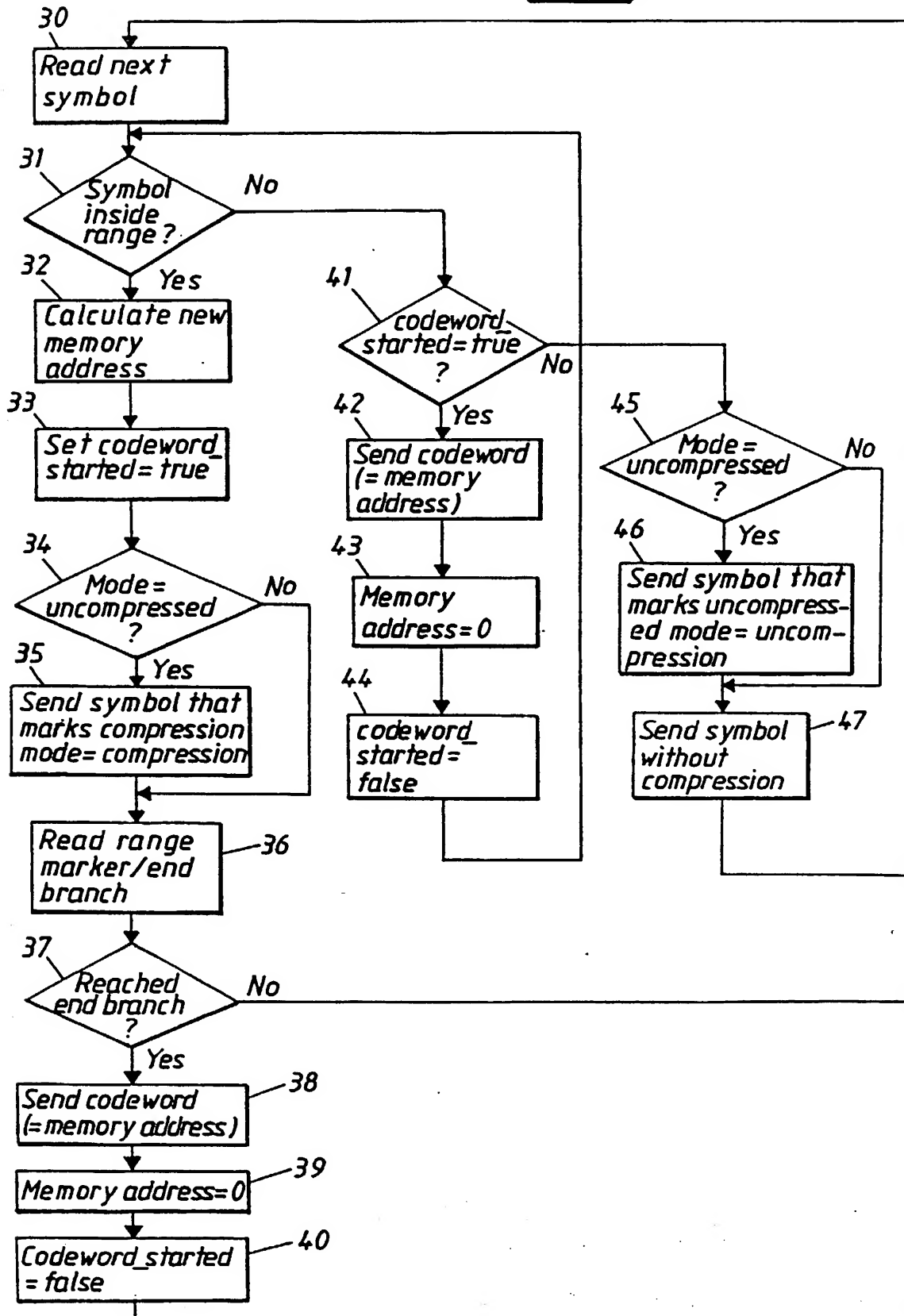


Fig. 11a

6/6

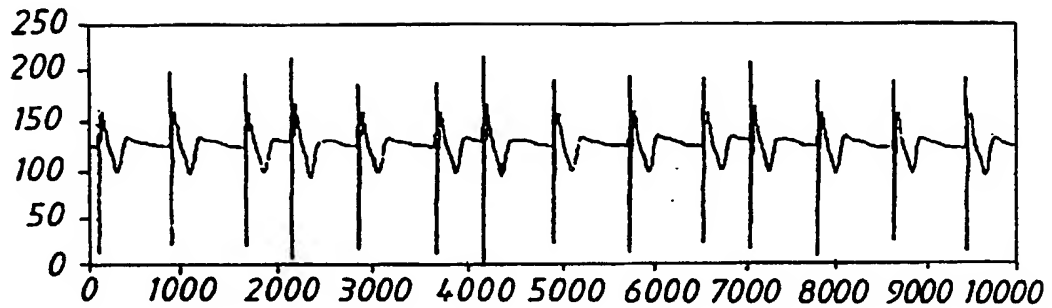


Fig. 11b

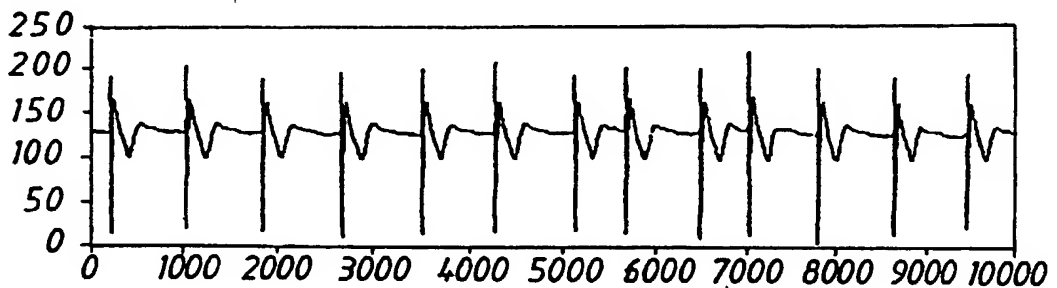


Fig. 11c

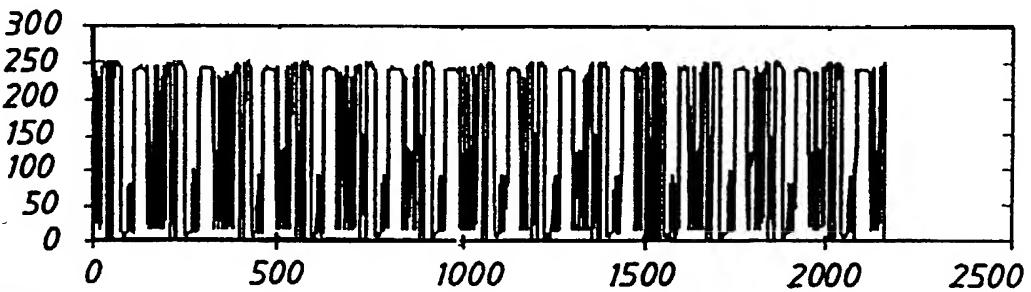


Fig. 11d

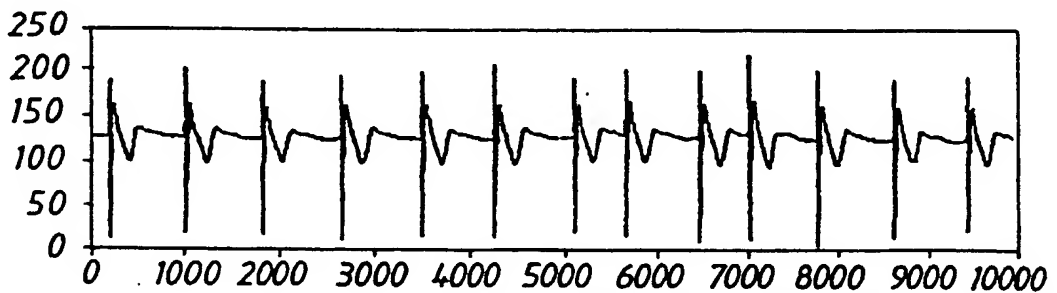
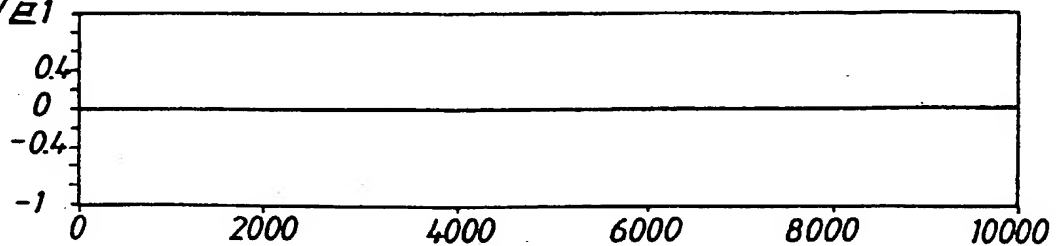


Fig. 11e1



BOX PCT
IN THE UNITED STATES DESIGNATED OFFICE
OF THE UNITED STATES PATENT AND TRADEMARK OFFICE
UNDER THE PATENT COOPERATION TREATY-CHAPTER II

APPLICANT:	Johan Lidman
ATTORNEY DOCKET NO.	P02,0086
INTERNATIONAL APPLICATION NO:	PCT/SE00/01744
INTERNATIONAL FILING DATE:	September 7, 2000

10 INVENTION: "COMPRESSION AND DECOMPRESSION CODING
SCHEME AND APPARATUS"

Assistant Commissioner for Patents,
Washington, D.C.

S I R:

15 Applicant herewith requests approval of the drawing changes in each of Figures 1, 2 and 8, as well as approval of further changes on each of sheets 1 through 6, as shown on the copies of sheets 1 through 6 marked in red attached hereto.

Submitted by,
Stan W. Noll (Reg. 28,982)

20 _____ SCHIFF, HARDIN & WAITE
CUSTOMER NO. 26574
 Patent Department
 6600 Sears Tower
 25 233 South Wacker Drive
 Chicago, Illinois 60606
 Telephone: 312/258-5790
 Attorneys for Applicant.

1 / 6

Fig. 1
(PRIOR ART)

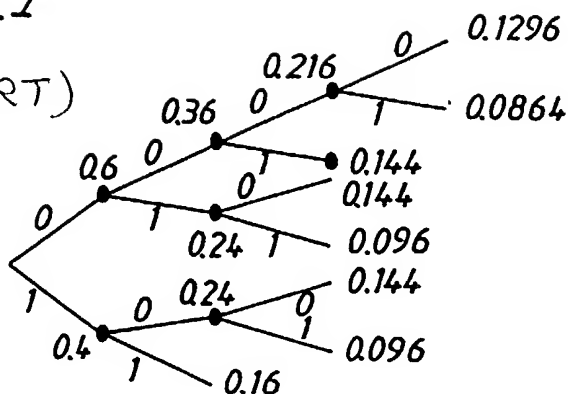


Fig. 2
(PRIOR ART)

Symbol sequences	Codeword	Codeword binary form
0000	1	001
0001	2	010
001	3	011
010	4	100
011	5	101
100	6	110
101	7	111
11	8	000

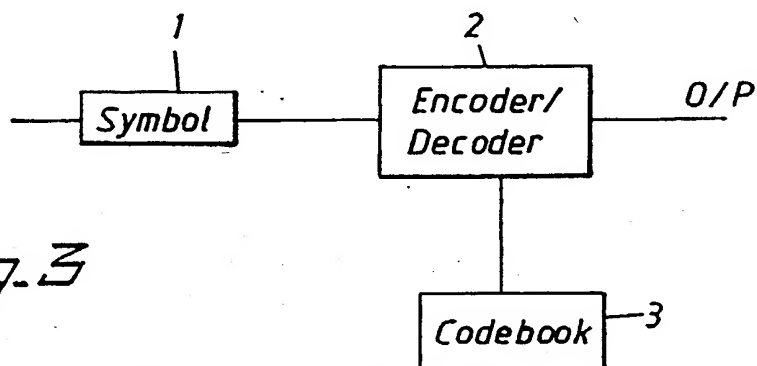


Fig. 3

2/6

Fig. 4a

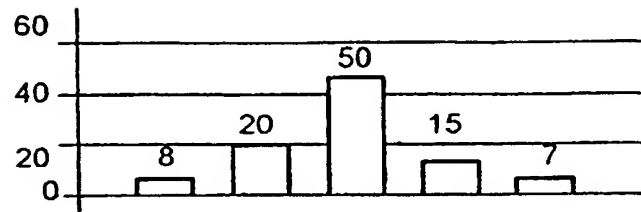


Fig. 4b

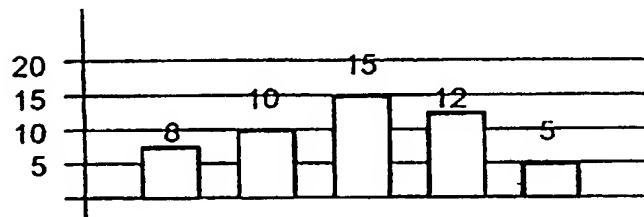


Fig. 4c

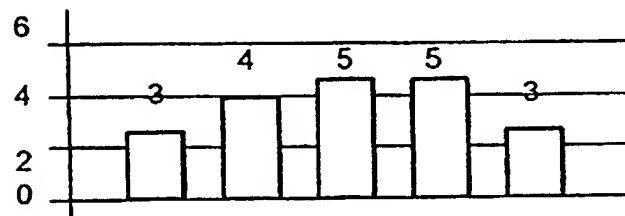


Fig. 4d

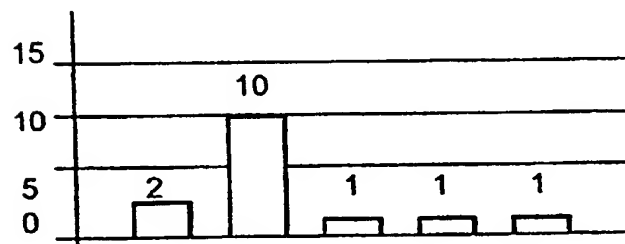
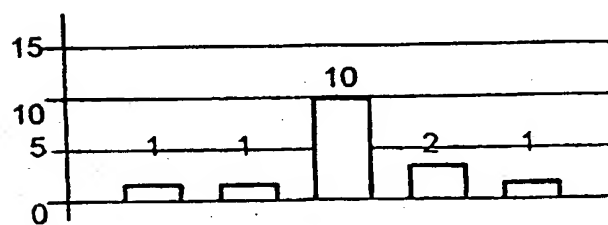


Fig. 4e



3 / 6

Fig. 5

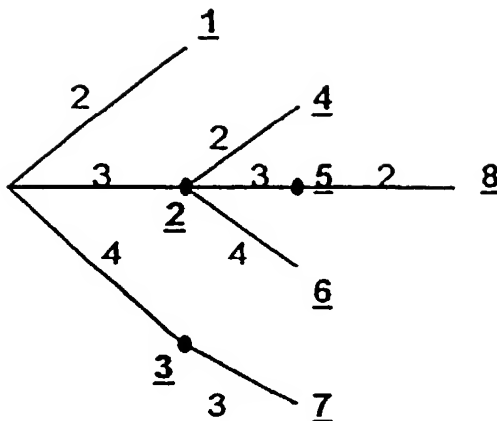
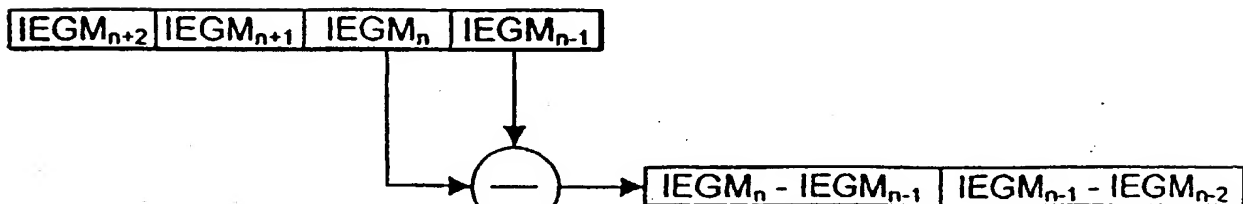


Fig. 6

Symbol sequence	Codeword
2	1
3	2
4	3
32	4
33	5
34	6
43	7
332	8

Fig. 7



4 / 6

Fig. 8

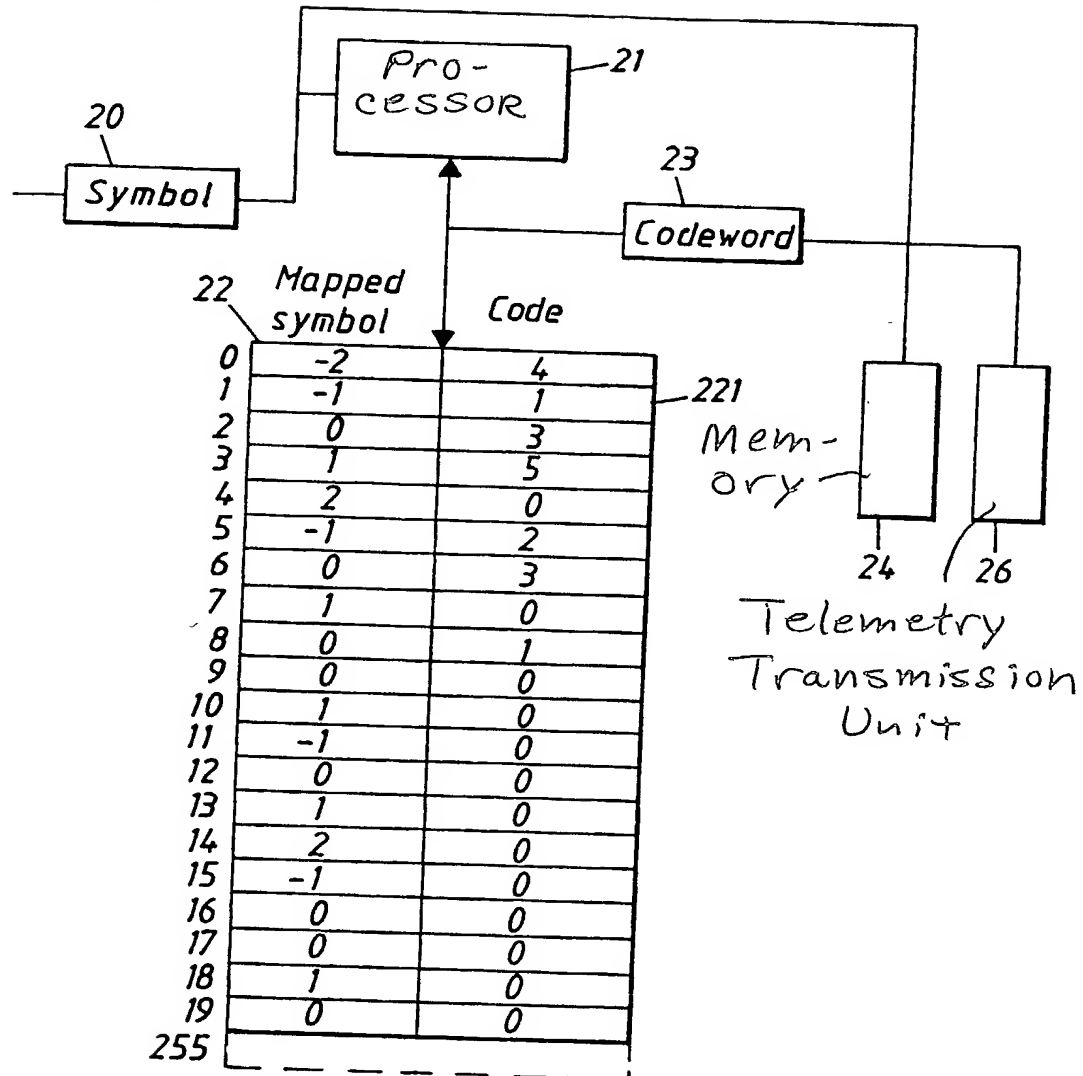


Fig. 9

25

Code No.	Possible branches	No. of branches
0	End	0
1	0	1
2	-1, 0	2
3	0, 1	2
4	-1, 0, 1	3
5	-1, 0, 1, 2	4
6	-2, -1, 0, 1	4
7	-2, -1, 0, 1, 2	5

5 / 6

Fig. 10

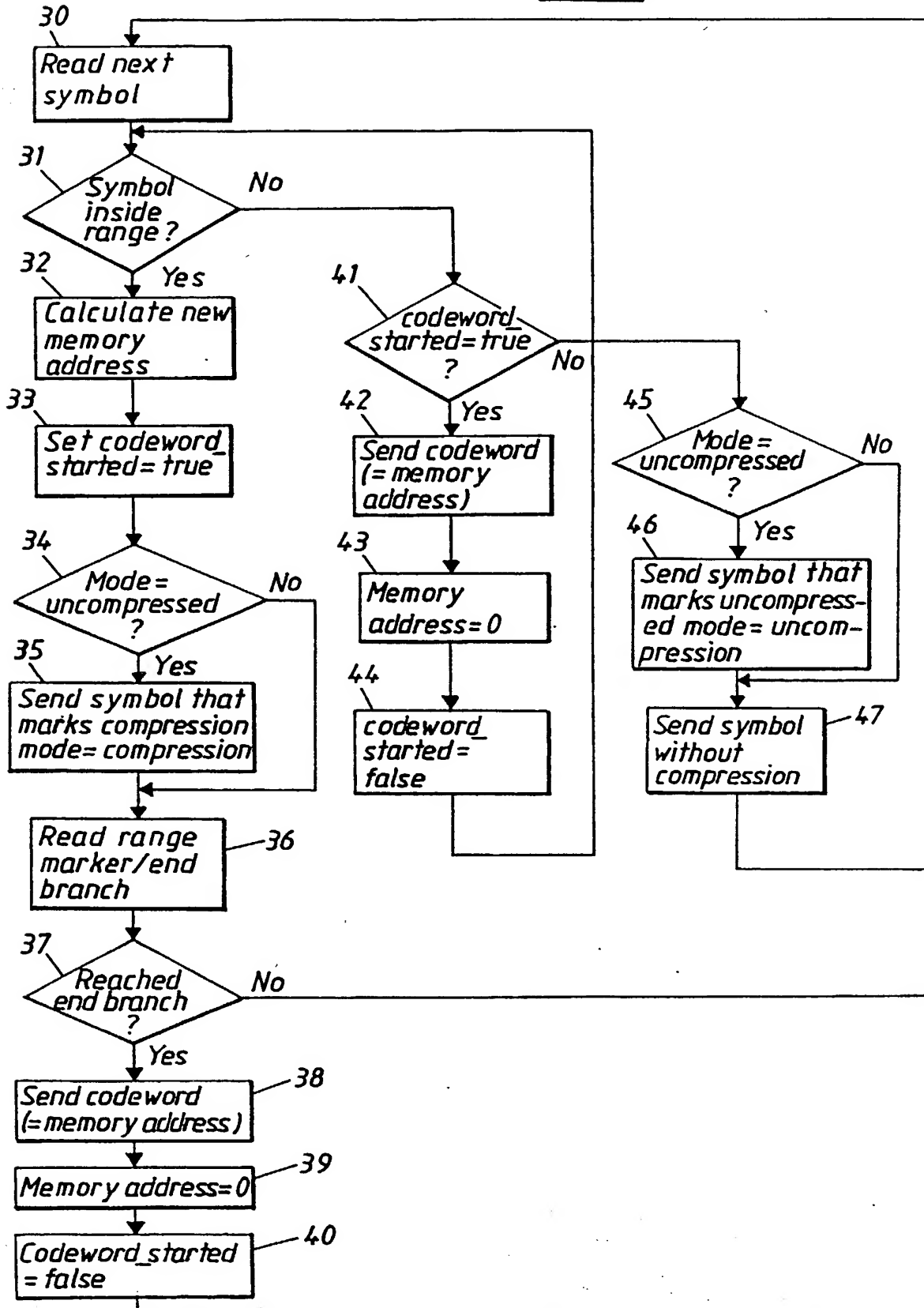


Fig. 11a

616

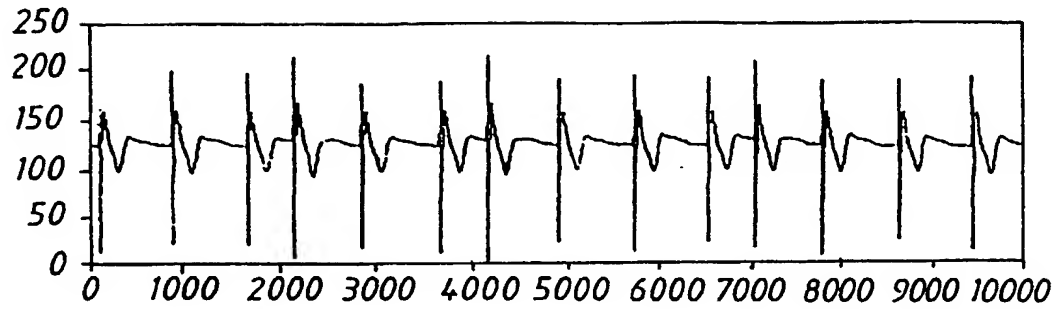


Fig. 11b

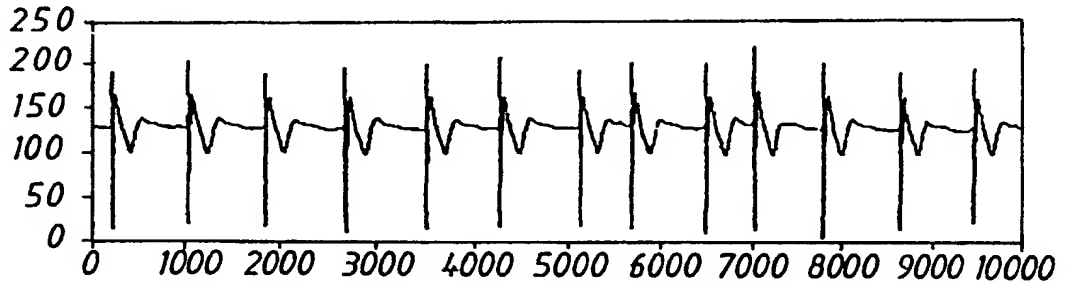


Fig. 11c

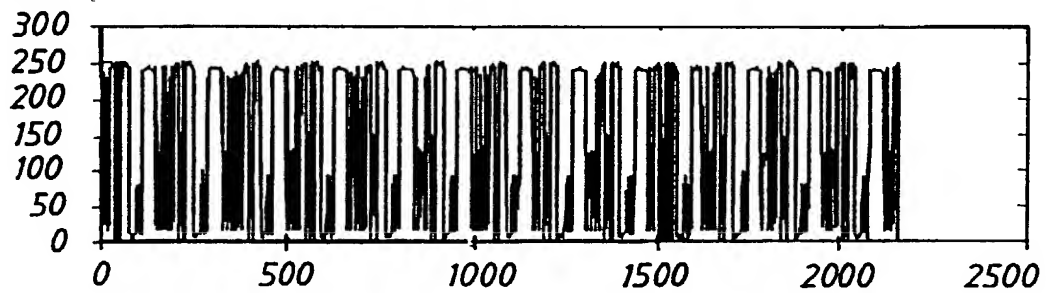


Fig. 11d

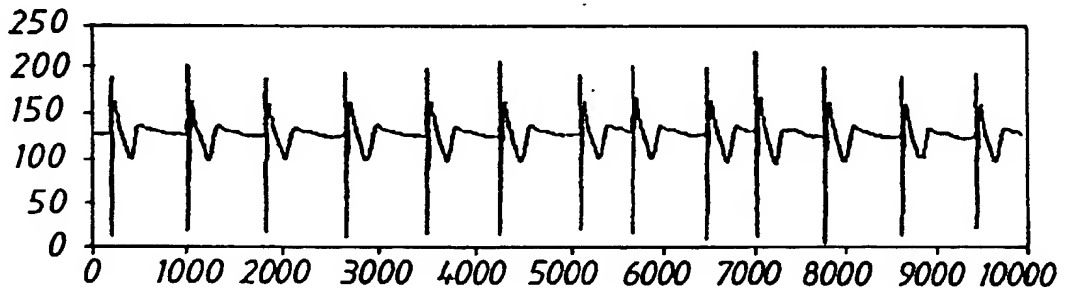
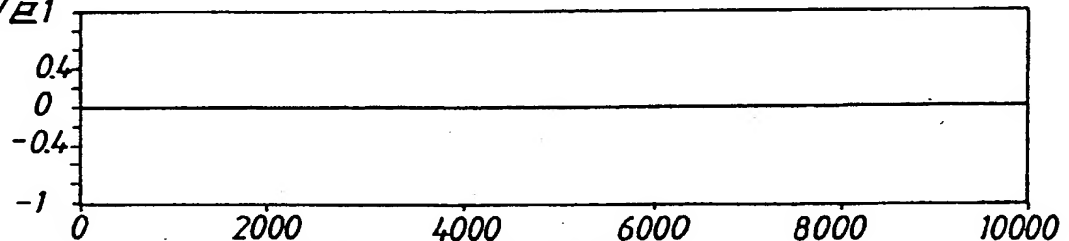


Fig. 11e



SUBSTITUTE SPECIFICATION

SPECIFICATION

TITLE

**“COMPRESSION AND DECOMPRESSION CODING SCHEME
AND APPARATUS”**

5 **BACKGROUND OF THE INVENTION**

Field of the Invention

The present invention relates generally to a coding scheme and apparatus for the compression and decompression of data. It is particularly directed to the compression of signals that exhibit so-called memory, where
10 a portion of a signal depends on the value of a preceding portion. The invention has particular application to medical systems, such as implantable pacemaker devices, which have limited memory but require the storage of large quantities of data.

Description of the Prior Art

15 Medical systems for monitoring physiological functions are becoming more complex as the need for diagnostic applications increases. In particular there is a need for intracardial detection systems and pacemaker systems capable of storing ever increasing numbers of signals, such as electrocardiogram signals (EGG and IEGM), pressure signals and
20 bioimpedance signals or the like, of ever increasing length. However, the available memory space for data storage is often restricted, particularly in implanted pacemaker systems. Perhaps more importantly in implanted systems, the amount of data that may be collected is also restricted by the transmission capacity of a telemetry link between an implanted device and

-2- **SUBSTITUTE SPECIFICATION**

its programmer or other external control device. For example, a defibrillator today typically requires a transmission time of up to 40 minutes for downloading to its controller all the data that can be collected. If the required quantities of data are to be made available for processing, this data must be
5 compressed.

Data compression can generally be divided into two forms. A first form is based on viewing a signal as a mathematical function and observing and utilizing characteristics of this function to compress data. The second form utilizes coding theory and is based on the statistics of multiple discrete
10 signal levels, or symbols, occurring in a signal.

A conventional algorithm working according to this latter principle is the Tunstall code. This code maps variable-length symbol segments of an input signal into fixed-length codewords. Since the codeword length is fixed, the number of codewords n is known in advance. The object is to assign
15 codewords to symbol segments that occur with approximately equal probability. The procedure begins with a set of symbol segments consisting of each of the individual symbols occurring in the input signal, such as m symbols in total. The most probable symbol is then removed from the set and replaced by m new segments, each of which is the removed symbol
20 suffixed by one of the m input symbols. This procedure continues until the number of symbol segments, M , in the set is equal to the number of codewords, n . The codewords are then assigned to the symbol segments.

-3- SUBSTITUTE SPECIFICATION

An example of Tunstall encoding is illustrated in Figs. 1 and 2. Fig. 1 shows a probability tree comprising nodes, and branches emanating from each node. In the illustrated example it is assumed that S codewords are available for compressing data, 1 to 8. The tree in Fig. 1 assumes that a source signal comprises two signals, '0' and '1'. Each symbol is represented by a first branch. The probability a symbol occurring is given at the node terminating the associated branch. Hence a '0' occurs with a probability of 0.6 and a '1' occurs with a probability of 0.4. The branch with the highest probability is expanded further by adding a second series of branches for each possible symbol. The '0' branch is thus bifurcated into a second '0' branch with a total probability of 0.36 and a '1' branch with probability 0.24. The first branch for symbol '1' now has the highest probability and is expanded in turn, resulting in two further branches with probability of 0.24 and 0.16. This process continues until the number of end branches equals the number of codewords, in the present example 8, with the probability of each branch occurring being relatively close. Each end branch represents a sequence of symbols. These sequences are assigned a codeword as shown in Fig. 2.

A problem associated with Tunstall encoding is that signals containing a large number of different symbols, for instance a large number of discrete signal levels, require a very large number of codewords. For example, in a pacemaker system, a typical electrogram or IEGM signal is represented by S bits sampled at 512 Hz. The number of symbols in this signal is thus 256.

-4- **SUBSTITUTE SPECIFICATION**

A Tunstall codebook for this signal would have to contain at least 256 codewords just to cover the different individual symbols. In order to obtain compression of the signal, the branches must be expanded further which adds a further 256 codewords per branch. The Tunstall code is a general-
5 purpose data compression algorithm and is not adapted to special classes of data. In particular it is not effective when applied to signals which exhibit memory. Typically, many of the signals monitored by medical systems, and cardiac pacers in particular, exhibit some memory, one example being the IEGM signal. The use of Tunstall encoding for processing sampled data in
10 medical systems, and specifically implanted systems, is thus limited.

SUMMARY OF THE INVENTION

It is an object of the present invention to provide a coding method and apparatus that allows a signal exhibiting memory to be compressed and decompressed efficiently and without distortion.

15 The above object is achieved in accordance with the present invention in a data coding method for converting a signal containing a plurality of symbols into codeword, and an apparatus and software product for implementing the method, including the steps of: establishing a set of codewords, monitoring a data signal and determining the most frequently
20 occurring symbols and/or sequences of symbols containing at least two symbols, allocating one codeword to each of the most frequently occurring of said symbols and/or symbol sequences.

-5- **SUBSTITUTE SPECIFICATION**

According to a further aspect, the invention provides a data compression method for compressing a data signal containing a plurality of symbols, including: converting the most frequently occurring symbols and/or symbol sequences into codewords, supplementing the remaining symbols with at least one codeword indicative of no compression.

The invention further proposes an arrangement for compressing and decompressing a data signal containing a plurality of symbols, including: means for storing codewords corresponding to symbols and/or symbol sequences, and means for determining if a symbol in said data signal corresponds to at least one codeword in the storage means and, when a symbol corresponds to only one codeword, for transmitting said codeword, wherein the determining means are further adapted to transmit a symbol if it corresponds to no codeword in the storage means.

According to a fourth aspect, the invention proposes a computer program product for converting a signal containing a plurality of symbols into a compressed signal, including computer readable program code means for establishing a set of codewords, determining the most frequently occurring symbols and/or sequences of symbols containing at least two symbols in a data signal and allocating one codeword to each of the most frequently occurring of said symbols and/or symbol sequences.

By providing codewords to only the most frequently occurring symbols and symbols sequences, the number of codewords required is greatly reduced. At the same time however the efficiency of the compression is

-6- **SUBSTITUTE SPECIFICATION**

increased, as these codewords are allocated to the varying length symbol sequences that appear with the highest frequency in the signal. This compression technique furthermore fully exploits any memory in a signal. A further difference over prior art coding schemes and the Tunstall code in particular is that codewords may be allocated to every node and end branch in a coding tree and not just the end branches. This is also particularly useful when compressing signals that exhibit memory such as signals monitoring physiological quantities such as the heartbeat, respiration rate and the like.

10 **DESCRIPTION OF THE DRAWINGS**

Fig. 1, as noted above, depicts a coding tree illustrating the Tunstall encoding algorithm.

Fig. 2, as noted above, depicts a coding table corresponding to the coding tree of Fig 1.

15 Fig. 3 is a schematic block diagram of an encoder/decoder (compressor/decompressor) according to the present invention.

Figs. 4a through 4e are a series of histograms illustrating the generation of a codebook according to the present invention.

20 Fig. 5 depicts a coding tree corresponding to the histograms of Figs. 4a through 4e.

Fig. 6 depicts a codebook corresponding to the coding tree of Fig. 5.

Fig. 7 is a schematic illustration of a pre-processing function in accordance with the present invention.

SECRET

THE UNITED STATES OF AMERICA

In a preferred embodiment of the present invention, the compression coding scheme according to the invention is used to compress an IEGM signal comprising 255 symbols ranging from 0 to 254. After pre-processing in accordance with the function of Fig. 7, the signal contains 509 possible symbols, ranging from -254 to +254. However, while the number of possible symbols has almost doubled, the form of the original IEGM signal is such that the processed signal contains mainly symbols close to 0 such as 1, -1,

11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 259 260 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350 351 352 353 354 355 356 357 358 359 360 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377 378 379 380 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395 396 397 398 399 400 401 402 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419 420 421 422 423 424 425 426 427 428 429 430 431 432 433 434 435 436 437 438 439 440 441 442 443 444 445 446 447 448 449 450 451 452 453 454 455 456 457 458 459 460 461 462 463 464 465 466 467 468 469 470 471 472 473 474 475 476 477 478 479 480 481 482 483 484 485 486 487 488 489 490 491 492 493 494 495 496 497 498 499 500 501 502 503 504 505 506 507 508 509 510 511 512 513 514 515 516 517 518 519 520 521 522 523 524 525 526 527 528 529 530 531 532 533 534 535 536 537 538 539 540 541 542 543 544 545 546 547 548 549 550 551 552 553 554 555 556 557 558 559 560 561 562 563 564 565 566 567 568 569 570 571 572 573 574 575 576 577 578 579 580 581 582 583 584 585 586 587 588 589 590 591 592 593 594 595 596 597 598 599 600 601 602 603 604 605 606 607 608 609 610 611 612 613 614 615 616 617 618 619 620 621 622 623 624 625 626 627 628 629 630 631 632 633 634 635 636 637 638 639 640 641 642 643 644 645 646 647 648 649 650 651 652 653 654 655 656 657 658 659 660 661 662 663 664 665 666 667 668 669 670 671 672 673 674 675 676 677 678 679 680 681 682 683 684 685 686 687 688 689 690 691 692 693 694 695 696 697 698 699 700 701 702 703 704 705 706 707 708 709 710 711 712 713 714 715 716 717 718 719 720 721 722 723 724 725 726 727 728 729 730 731 732 733 734 735 736 737 738 739 740 741 742 743 744 745 746 747 748 749 750 751 752 753 754 755 756 757 758 759 760 761 762 763 764 765 766 767 768 769 770 771 772 773 774 775 776 777 778 779 780 781 782 783 784 785 786 787 788 789 790 791 792 793 794 795 796 797 798 799 800 801 802 803 804 805 806 807 808 809 810 811 812 813 814 815 816 817 818 819 820 821 822 823 824 825 826 827 828 829 830 831 832 833 834 835 836 837 838 839 840 841 842 843 844 845 846 847 848 849 850 851 852 853 854 855 856 857 858 859 860 861 862 863 864 865 866 867 868 869 870 871 872 873 874 875 876 877 878 879 880 881 882 883 884 885 886 887 888 889 890 891 892 893 894 895 896 897 898 899 900 901 902 903 904 905 906 907 908 909 910 911 912 913 914 915 916 917 918 919 920 921 922 923 924 925 926 927 928 929 930 931 932 933 934 935 936 937 938 939 940 941 942 943 944 945 946 947 948 949 950 951 952 953 954 955 956 957 958 959 960 961 962 963 964 965 966 967 968 969 970 971 972 973 974 975 976 977 978 979 980 981 982 983 984 985 986 987 988 989 990 991 992 993 994 995 996 997 998 999 1000 1001 1002 1003 1004 1005 1006 1007 1008 1009 1010 1011 1012 1013 1014 1015 1016 1017 1018 1019 1020 1021 1022 1023 1024 1025 1026 1027 1028 1029 1030 1031 1032 1033 1034 1035 1036 1037 1038 1039 1040 1041 1042 1043 1044 10

-12- **SUBSTITUTE SPECIFICATION**

class of signal, so that one codebook is used for several patients. A further option would be to utilize an adaptive procedure, whereby the statistics of the signal are observed and a codebook newly generated prior to each compression. This codebook would then be optimized for each signal generated in a pacemaker. However, it will be understood that the codebook must be retained long enough to enable the compressed data to be decompressed.

In the coding procedures described above, a codeword is allocated to every node and end branch of a probability tree. It will, however, be understood that codewords could be assigned only to the end branches of the tree. The efficiency of such a code would depend on the characteristics of the starting signal.

An arrangement for encoding and decoding a signal is illustrated in Fig. 8. The arrangement includes an input stage 20 for reading a symbol. A processor 21, that preferably takes the form of a single chip microprocessor with associated memory, is coupled to the input stage. A codebook memory 22 having 256 8-bit memory locations with addresses from 0 to 255 is connected to the processor 21. An output stage 23 for the codewords is coupled to the memory 22 and processor 21. This output stage 23 is finally coupled to a storage memory 24 for storing the compressed data and to a telemetry transmission unit 26, which forwards the data to a remote external programmer or controller. The input stage 20 is also coupled to the storage memory 24 and telemetry transmission unit 26

-13- **SUBSTITUTE SPECIFICATION**

for transmitting uncompressed data. The encoding and decoding arrangement is preferably preceded by a preprocessing stage as shown in Fig. 7.

The function of this arrangement is basically as follows. A first symbol
5 is read by the input stage 20. The processor 21 checks whether this symbol corresponds to more than one symbol sequence in the codeword memory 22 and if so, the next symbol is read. This process is repeated until the sequence of symbols read corresponds to only one codeword. This codeword is then emitted in place of the symbol sequence and is either
10 stored in the memory 24 or sent directly to an external device via the telemetry transmission unit 26. If a symbol read corresponds to no coded sequences it is transmitted unchanged to the storage memory 24 or transmission unit 26 but preceded by the codeword 254 indicating uncompressed data.

15 The addresses 0 to 255 of the codebook memory 22 are the codewords. The processor 21 furthermore performs a mapping between the incoming symbols that form at least part of a coded sequence and each memory cell 221. This is illustrated in more detail in Fig. 9. The first symbol in a sequence will cause the processor 21 to access the first memory cell, in
20 ascending order of address, to which the symbol is mapped. These first memory cells 221 correspond to the first branches in a probability tree. Thus if the first symbol in a sequence is '1', the processor 21 will access the fourth memory location (address 3), since this is the first location 221 to which a '1'

is mapped. Each memory cell 221 further contains information indicating whether the mapped symbol corresponds to more than one coded symbol sequence. This information is shown on the right hand side of each memory cell 221. The information is in the form of a code represented by the numbers 0 to 7, which indicate both the number of possible further branches and the following symbols corresponding to the further branches. A conversion table 25 shows the significance of each number. Hence it can be seen that '0' indicates an end branch, '2' represents two possible further branches with the following symbols '1' and '2'. Any symbol mapped to a memory cell 221 containing a '0' will result in the processor 21 transmitting the address of the memory cell as a codeword. Any other number contained in the memory cell 221 indicates that at least one further branch is possible. The processor 21 then fetches the next symbol from the input stage 20 and determines whether this symbol forms part of the possible codewords. If it does, the processor 21 calculates the address of the next memory cell 221 and the process is repeated.

The next address is calculated by summing the number of possible branches contained in all addresses starting from the first address up to the present address, and then adding the position of the fetched symbol in the list of possible symbols given in the conversion table 25. The sum of possible branches is equal to the sum of the first mapped memory locations representing the initial branches of the probability tree and the possible branches stored in all previous memory locations. Thus an input sequence

-15- SUBSTITUTE SPECIFICATION

consisting of 1, 2 would result in a first mapping being made to memory location 3. This contains the code 5, which indicates that four further branches are possible. The next symbol is fetched. It is verified that '2' corresponds to one of the possible branches. The symbol '2' has the fourth position in the list of branches (-1, 0, 1, 2) given in the conversion table 25. Thus the sequence 1, 2 is a valid coded sequence. The next address is equal to the sum of the initial basic branches (i.e. the addresses to which a first symbol can be mapped), which in the present example is 5 (addresses 0 to 4). To this is added the sum of the branches contained in memory locations up to address 3, and the position of the subsequent symbol in the list indicated in memory location 3. This gives a total of $5 + (3 + 1 + 2) + 4 = 15$. The next address is thus the 15th location or address 14, since the addresses start with 0. This is verified by the mapping of the symbol 2 indicated in the left column of the cell with address 14.

15 The conversion table 25 alternatively may be more complex in structure and provide absolute memory locations corresponding to each possible branch. In this way the next address would not need to be calculated, but more storage capacity would be needed.

The conversion table is preferably stored in the processor 21. However, it may be possible to store some of the information about the branches in the codebook memory 22, depending on how much capacity is available for each address.

-16- **SUBSTITUTE SPECIFICATION**

The arrangement illustrated in Fig. 8 is a schematic representation of possible encoding and decoding hardware. It will however, be understood that the functions of the various elements shown in Fig. 8 may be performed entirely in a digital processor system operating under the control of a program. The codebook memory 22 could then be implemented virtually
5 from part of the memory space incorporated in the processor system.

Fig. 10 is a flow chart illustrating the procedure for compressing data using the arrangement shown in Figs. 8 and 9. The procedure starts at step 30 with the reading of a symbol. In step 31 it is determined whether the
10 symbol is inside the designated range, i.e. whether the symbol forms part of a coded sequence and can be mapped to a memory location. If this is the case, the process moves to step 32 and the new memory address is calculated. In the following step 33 a marker, 'codeword_started', indicating that the coding of a sequence has started, is set. In step 34 it is verified
15 whether the data last sent was uncompressed. If this is true, in step 35 the symbol indicating compression is sent. If the last data sent was in compressed mode, the process moves directly from step 34 to step 36, where the contents of the memory location are read and it is determined whether an end branch has been reached. If the end branch has not been
20 reached the procedure returns to step 30 and the next symbol is fetched. If the end branch is reached, the procedure passes to step 38, where the memory address is sent as the codeword. In step 39, the memory address is reset to 0 and in step 40, the marker 'codeword_started' is reset to false,

-17- **SUBSTITUTE SPECIFICATION**

because the coding of a symbol or symbol sequence is terminated. The process then returns to step 30, and the next symbol fetched. If a symbol is discovered to be out of range in step 31, indicating that the symbol does not form part of a coded sequence, the procedure goes to step 41 where the status of the marker 'codeword_started' is verified. If this is true, this means that a codeword has been started, but the subsequent symbol does not form part of the coded sequence. In step 42, therefore, the memory address is sent as the codeword. The memory address is then reset to 0 in step 43, the marker 'codeword_started' reset to false in step 44 and the procedure returns to step 31, where the fetched symbol is checked against the starting symbols of coded sequences to verify if it forms part of a coded sequence. If in step 41, it is determined that no codeword has been started, i.e. the status of the 'codeword_started' marker is false, this means that the fetched symbol is not contained in any coded sequence. In step 45, the transmission mode is checked. If the last data sent was compressed, the symbol indicating uncompressed data is sent in step 46 followed by the read symbol in step 47. If the last transmission was not in compression mode, the symbol is sent in step 47. The procedure then returns to the start at step 30.

Decompression is the exact reverse of the compression procedure described above. Each codeword is converted into the corresponding symbol or symbol sequence. The symbols are then summed to retrieve the original uncompressed data.

-18- **SUBSTITUTE SPECIFICATION**

Figs. 11a to 11e illustrate an example using the coding algorithm described above. The algorithm was first trained, i.e. the codebook generated, using an IEGM signal containing 10000 samples sampled at 512 Hz with 8-bit resolution. The training signal is illustrated in Fig. 11a. The
5 algorithm was then tested on an IEGM signal containing 10000 samples sampled at 512 Hz with 8-bit resolution. This uncompressed test signal is depicted in Fig. 11b. Fig. 11c shows the signal after compression. This signal contains 2149 samples which gives a compression ratio of 4.6. Fig. 11d shows the signal of Fig. 11c after decompression. Finally, Fig. 11e
10 shows the difference signal between the original signal depicted in Fig. 11b and the decompressed signal of Fig. 11d. The signal is entirely free of distortion.

In the coding procedures described above, a codeword is allocated to every node and end branch of a probability tree. It will, however, be
15 understood that codewords could be assigned only to the end branches of the tree.

Although modifications and changes may be suggested by those skilled in the art, it is the intention of the inventor to embody within the patent
warranted hereon all changes and modifications as reasonably and properly
20 come within the scope of his contribution to the art.

MARKED-UP COPY

SPECIFICATION

TITLE

**“COMPRESSION AND DECOMPRESSION CODING SCHEME
AND APPARATUS”**

5 BACKGROUND OF THE INVENTION

Field of the Invention

[Field of invention]

The present invention relates generally to a coding scheme and apparatus for the compression and decompression of data. It is particularly
10 directed to the compression of signals that exhibit so-called memory, where a portion of a signal depends on the value of a preceding portion. The invention has particular application to medical systems, such as implantable pacemaker devices, which have limited memory but require the storage of large quantities of data.

15 Description of the Prior Art

[Background art]

Medical systems for [the] monitoring [of] physiological functions are becoming more complex as the need for diagnostic applications increases. In particular there is a need for intracardial detection systems and
20 pacemaker systems capable of storing ever increasing numbers of signals, such as [electrocardiogram] electrocardiogram signals (EGG and IEGM), pressure signals and bioimpedence signals or the like, of ever increasing length. However, the available memory space for data storage is often restricted, particularly in implanted pacemaker systems. Perhaps more

-2-

MARKED-UP COPY

importantly in implanted systems, the amount of data that may be collected is also restricted by the transmission capacity of a telemetry link between an implanted device and its programmer or other external control device. For example, a defibrillator today typically requires a transmission time of up to
5 40 minutes for downloading to its controller all the data that can be collected. If the required quantities of data are to be made available for processing, this data must be compressed.

Data compression can generally be divided into two forms. A first form is based on viewing a signal as a mathematical function and observing
10 and [utilising] utilizing characteristics of this function to compress data. The second form [utilises] utilizes coding theory and is based on the statistics of multiple discrete signal levels, or symbols, occurring in a signal.

A conventional algorithm working according to this latter principle is the Tunstall code. This code maps variable-length symbol segments of an
15 input signal into fixed-length codewords. Since the codeword length is fixed, the number of codewords n is known in advance. The object is to assign codewords to symbol segments that occur with approximately equal probability. The procedure begins with a set of symbol segments consisting of each of the individual symbols occurring in the input signal, [say] such as
20 m symbols in [all] total. The most probable symbol is then removed from the set and replaced by m new segments, each of which is the removed symbol suffixed by one of the m input symbols. This procedure continues until the

number of symbol segments, Manager, in the set is equal to the number of codewords, n . The codewords are then assigned to the symbol segments.

An example of Tunstall encoding is illustrated in Figs. 1 and 2. Fig. 1 shows a probability tree comprising nodes, and branches emanating from each node. In the illustrated example it is assumed that S codewords are available for compressing data, 1 to 8. The tree in Fig. 1 assumes that a source signal comprises two signals, '0' and '1'. Each symbol is represented by a first branch. The probability a symbol occurring is given at the node terminating the associated branch. Hence a '0' occurs with a probability of 0.6 and a '1' occurs with a probability of 0.4. The branch with the highest probability is expanded further by adding a second series of branches for each possible symbol. The '0' branch is thus bifurcated into a second '0' branch with a total probability of 0.36 and a '1' branch with probability 0.24. The first branch for symbol '1' now has the highest probability and is expanded in turn, resulting in two further branches with probability of 0.24 and 0.16. This process continues until the number of end branches equals the number of codewords, in the present example 8, with the probability of each branch occurring being relatively close. Each end branch represents a sequence of symbols. These sequences are assigned a codeword as shown in Fig. 2.

A problem associated with Tunstall encoding is that signals containing a large number of different symbols, for instance a large number of discrete signal levels, require a very large number of codewords. For example, in a

data signal and allocating one codeword to each of the most frequently occurring of said symbols and/or symbol sequences.

By providing codewords to only the most frequently occurring symbols and symbols sequences, the number of codewords required is greatly reduced. At the same time however the efficiency of the compression is increased, as these codewords are allocated to the varying length symbol sequences that appear with the highest frequency in the signal. This compression technique furthermore fully exploits any memory in a signal. A further difference over prior art coding schemes and the Tunstall code in particular is that codewords may be allocated to every node and end branch in a coding tree and not just the end branches. This is also particularly useful when compressing signals that exhibit memory such as signals monitoring physiological quantities such as the heartbeat, respiration rate and the like.

15 [BRIEF DESCRIPTION OF THE DRAWINGS

Further objects and advantages of the present invention will become apparent from the following description of the preferred embodiments that are given by way of example with reference to the accompanying drawings, in which:

20 Fig. 1 depicts a coding tree illustrating the Tunstall encoding algorithm;

Fig. 2 depicts a coding table corresponding to the coding tree of Fig.

- Fig. 3 depicts a schematic block diagram of an encoder/decoder (compressor/decompressor) according to the present invention.
- Figs. 4a depicts a series of histograms illustrating the generation of a codebook according to the present invention;
- To 4e
- 5 Fig. 5 depicts a coding tree corresponding to the histograms of Fig. 3;
- Fig. 6 depicts a codebook corresponding to the tree of Fig. 4;
- Fig. 7 is a schematic illustration of a pre-processing function in accordance with the present invention.
- 10 Fig. 8 is a schematic diagram showing an arrangement for compressing and decompressing data according to the present invention;
- Fig. 9 is a schematic illustration of a codebook memory illustrating the mapping between symbols and memory locations;
- 15 Fig. 10 is a flow chart illustrating a method for compressing data utilising the arrangement of Fig. 8; and
- Figs 11a to show a sequence of graphs illustrating the compression and 11e subsequent decompression of an IEGM signal.

DETAILED DESCRIPTION OF THE DRAWINGS]

20

DESCRIPTION OF THE DRAWINGS

Fig. 1, as noted above, depicts a coding tree illustrating the Tunstall encoding algorithm.

Fig. 2, as noted above, depicts a coding table corresponding to the coding tree of Fig 1.

Fig. 3 is a schematic block diagram of an encoder/decoder (compressor/decompressor) according to the present invention.

5 Figs. 4a through 4e are a series of histograms illustrating the generation of a codebook according to the present invention.

Fig. 5 depicts a coding tree corresponding to the histograms of Figs. 4a through 4e.

Fig. 6 depicts a codebook corresponding to the coding tree of Fig. 5.

10 Fig. 7 is a schematic illustration of a pre-processing function in accordance with the present invention.

Fig. 8 is a schematic diagram of an arrangement for compressing and decompressing data according to the present invention.

15 Fig. 9 is a schematic illustration of a codebook memory illustrating the mapping between symbols and memory locations in accordance with the invention.

Fig. 10 is a flowchart illustrating a method for compressing data utilizing the inventive arrangement of Fig. 8.

20 Figs. 11a through 11e are a sequence of graphs illustrating the compression and subsequent decompression of an IEGM signal, in accordance with the present invention.

The generation of a codebook according to the present invention commences with determining the number of codewords to be used. This is

selected as a function of the total number of symbols contained in an input signal and the degree of compression required.

The codebook is generated by observing the input signal during a test phase and determining the probabilities of the symbols and sequences of symbols occurring. This is accomplished by observing the input signal during a set time period, for example 20 s, and noting the symbol that occurs with the highest frequency during this period. The signal is then observed again for the same time period and the symbol that follows the noted symbol most frequently is noted. This process continues until the number of most frequently occurring symbols and/or symbol sequences noted is equal to the number of codewords in the codebook. It will be understood, that only those symbols or symbol sequences that occur most frequently will be coded. This allows the number of codewords to be kept to a reasonable number.

A simplified example of this process is illustrated in Figs. [4] 4a-4e, 5 and 6 with reference to the generation of a codebook containing 8 symbols. The input signal consists of 5 symbols 1, 2, 3, 4, 5.

Turning now to Fig. 4a, a first histogram is produced after observing the input signal for a set period. It is evident that the most frequently occurring symbol is 3 with an occurrence of 50. Fig. 4b shows a histogram produced after the same period of time and gives the frequency of the different symbols that follow 3. Thus the sequence 32 has occurred ten times, the sequence 33 fifteen times, and the sequence 34 twelve times. Fig. 4c shows a histogram of the symbols following 2 and Fig. 4d shows a

histogram of the symbols following the symbol sequence 3, 3. Finally Fig. 4e shows the histogram of the symbols following the symbol 4. The codebook is established by determining which of the 8 symbols and/or symbol sequences occurred most frequently. This is illustrated schematically in a code tree in Fig. 5. In this code tree, a codeword has been assigned to every node and end branch. Hence the symbol 3 has been allocated the codeword '2' but the symbol sequence 33 has also been allocated a codeword, '5' and the symbol sequence 332 has been allocated the codeword '8'. This is [summarised] summarized in tabular form in Fig. 6.

10 It is apparent from the above example that codewords are not assigned to every symbol of an input signal. Thus when these unassigned symbols are read by an encoder, the symbol is transmitted uncompressed. To distinguish the uncompressed data from the compressed data, it is [preferably] preferable to [utilise] utilize some form of distinguishing symbol or codeword. Thus at least one codeword will be reserved for transmitting uncoded input symbols.

20 Depending on the signal to be compressed, it may be desirable to pre-process the signal in order to increase the frequency of a very few symbols and sequences. This considerably increases the efficiency of the compression. For example, pre-processing can be very effective when compressing sampled IEGM, EGG or other physiological signals that monitor some form of periodic activity.

Fig. 7 schematically illustrates the preferred function of such a pre-processor. In Fig. 7 an input data stream denoted by 10 comprises the symbols IEGMn-1, IEGMn, IEGMn+1 and IEGMn+2. The function generates the difference symbol value between a symbol and a preceding symbol. The
5 output data stream 12 thus comprises the symbols (IEGMn - IEGMn-1) and (IEGMn-1 - IEGMn-2), etc.

In a preferred embodiment of the present invention, the compression coding scheme according to the invention is used to compress an IEGM signal comprising 255 symbols ranging from 0 to 254. After pre-processing
10 in accordance with the function of Fig. 7, the signal contains 509 possible symbols, ranging from -254 to +254. However, while the number of possible symbols has almost doubled, the form of the original IEGM signal is such that the processed signal contains mainly symbols close to 0 such as 1, -1, 2, -2 etc. Thus the concentration of a very few symbols and symbol
15 sequences is increased by the difference function.

A codebook generated for a pre-processed IEGM signal of the type described above can typically be efficiently compressed [utilising] utilizing a codebook containing 254 codewords, for example 254 8-bit words ranging from 0 to 253. The 254 most frequently occurring symbols and/or symbol
20 sequences are then converted into codewords. The 8-bit words with values 254 and 255 are then reserved. These are [utilised] utilized to signal whether data is compressed or uncompressed. Preferably, one reserved codeword, for example 255, is sent to indicate that uncompressed data

follows. The other codeword 254 can then be [utilised] utilized to signal that compressed data is following. To avoid having to generate different symbols for negative and positive symbols, an uncompressed negative symbol is preferably indicated by preceding it with both symbols sent contiguously, for
5 example, 255 followed by 254 followed by the uncompressed equivalent positive symbol.

The codebook is preferably generated for a class of signals and retained for different signals of the same class so that it does not need to be re-established prior to compressing data. For example, a cardiac pacer
10 containing data compression and decompression circuitry for processing some physiological measurement such as an IEGM signal, EGG signal or bioimpedance signal could be subjected to a training phase for each patient to establish a codebook that is specific to each patient. The training phase could also be performed using a test sequence that is representative of the
15 class of signal, so that one codebook is used for several patients. A further option would be to [utilise] utilize an adaptive procedure, whereby the statistics of the signal are observed and a codebook newly generated prior to each compression. This codebook would then be [optimised] optimized for each signal generated in a pacemaker. However, it will be understood
20 that the codebook must be retained long enough to enable the compressed data to be decompressed.

In the coding procedures described above, a codeword is allocated to every node and end branch of a probability tree. It will, however, be

understood that codewords could be assigned only to the end branches of the tree. The efficiency of such a code would depend on the characteristics of the starting signal.

An arrangement for encoding and decoding a signal is illustrated in Fig. 8. The arrangement includes an input stage 20 for reading a symbol. A processor 21, that preferably takes the form of a single chip microprocessor with associated memory, is coupled to the input stage. A codebook memory 22 having 256 8-bit memory locations with addresses from 0 to 255 is connected to the processor 21. An output stage 23 for the codewords is coupled to the memory 22 and processor 21. This output stage 23 is finally coupled to a storage memory 24 for storing the compressed data and to a telemetry transmission unit 26, which forwards the data to a remote external programmer or controller. The input stage 20 is also coupled to the storage memory 24 and telemetry transmission unit 26 for transmitting uncompressed data. The encoding and decoding arrangement is preferably preceded by a preprocessing stage as shown in Fig. 7.

The function of this arrangement is basically as follows. A first symbol is read by the input stage 20. The processor 21 checks whether this symbol corresponds to more than one symbol sequence in the codeword memory 22 and if so, the next symbol is read. This process is repeated until the sequence of symbols read corresponds to only one codeword. This codeword is then emitted in place of the symbol sequence and is either

stored in the memory 24 or sent directly to an external device via the
telemetry transmission unit 26. If a symbol read corresponds to no coded
sequences it is transmitted unchanged to the storage memory 24 or
transmission unit 26 but preceded by the codeword 254 indicating
5 uncompressed data.

The addresses 0 to 255 of the codebook memory 22 are the
codewords. The processor 21 furthermore performs a mapping between the
incoming symbols that form at least part of a coded sequence and each
memory cell 221. This is illustrated in more detail in Fig. 9. The first symbol
10 in a sequence will cause the processor 21 to access the first memory cell, in
ascending order of address, to which the symbol is mapped. These first
memory cells 221 correspond to the first branches in a probability tree. Thus
if the first symbol in a sequence is '1', the processor 21 will access the fourth
memory location (address 3), since this is the first location 221 to which a '1'
15 is mapped. Each memory cell 221 further contains information indicating
whether the mapped symbol corresponds to more than one coded symbol
sequence. This information is shown on the right hand side of each memory
cell 221. The information is in the form of a code represented by the
numbers 0 to 7, which indicate both the number of possible further branches
20 and the following symbols corresponding to the further branches. A
conversion table 25 shows the significance of each number. Hence it can
be seen that '0' indicates an end branch, '2' represents two possible further
branches with the following symbols '1' and '2'. Any symbol mapped to a

memory cell 221 containing a '0' will result in the processor 21 transmitting the address of the memory cell as a codeword. Any other number contained in the memory cell 221 indicates that at least one further branch is possible. The processor 21 then fetches the next symbol from the input stage 20 and
5 determines whether this symbol forms part of the possible codewords. If it does, the processor 21 calculates the address of the next memory cell 221 and the process is repeated.

The next address is calculated by summing the number of possible branches contained in all addresses starting from the first address up to the
10 present address, and then adding the position of the fetched symbol in the list of possible symbols given in the conversion table 25. The sum of possible branches is equal to the sum of the first mapped memory locations representing the initial branches of the probability tree and the possible branches stored in all previous memory locations. Thus an input sequence
15 consisting of 1, 2 would result in a first mapping being made to memory location 3. This contains the code 5, which indicates that four further branches are possible. The next symbol is fetched. It is verified that '2' corresponds to one of the possible branches. The symbol '2' has the fourth position in the list of branches (-1, 0, 1, 2) given in the conversion table 25.
20 Thus the sequence 1, 2 is a valid coded sequence. The next address is equal to the sum of the initial basic branches (i.e. the addresses to which a first symbol can be mapped), which in the present example is 5 (addresses 0 to 4). To this is added the sum of the branches contained in memory

locations up to address 3, and the position of the subsequent symbol in the list indicated in memory location 3. This gives a total of $5 + (3 + 1 + 2) + 4 = 15$. The next address is thus the 15th location or address 14, since the addresses start with 0. This is verified by the mapping of the symbol 2
5 indicated in the [left-hand] left column of the cell with address 14.

The conversion table 25 [may] alternatively may be more complex in structure and provide absolute memory locations corresponding to each possible branch. In this way the next address would not need to be calculated, but more storage capacity would be needed.

10 The conversion table is preferably stored in the processor 21. However, it may be possible to store some of the information about the branches in the [codeword] codebook memory 22, depending on how much capacity is available for each address.

15 The arrangement illustrated in Fig. 8 is a schematic representation of possible encoding and decoding hardware. It will however, be understood that the functions of the various elements shown in Fig. 8 may be performed entirely in a digital processor system operating under the control of a program. The codebook memory 22 could then be implemented virtually from part of the memory space incorporated in the processor system.

20 Fig. 10 is a flow chart illustrating the procedure for compressing data using the arrangement shown in Figs. 8 and 9. The procedure starts at step 30 with the reading of a symbol. In step 31 it is determined whether the symbol is inside the designated range, i.e. whether the symbol forms part of

a coded sequence and can be mapped to a memory location. If this is the case, the process moves to step 32 and the new memory address is calculated. In the following step 33 a marker, 'codeword_started', indicating that the coding of a sequence has started, is set. In step 34 it is verified

5 whether the data last sent was uncompressed. If this is true, in step 35 the symbol indicating compression is sent. If the last data sent was in compressed mode, the process moves directly from step 34 to step 36, where the contents of the memory location are read and it is determined whether an end branch has been reached. If the end branch has not been

10 reached the procedure returns to step 30 and the next symbol is fetched. If the end branch is reached, the procedure passes to step 38, where the memory address is sent as the codeword. In step 39, the memory address is reset to 0 and in step 40, the marker 'codeword_started' is reset to false, because the coding of a symbol or symbol sequence is terminated. The

15 process then returns to step 30, and the next symbol fetched. If a symbol is discovered to be out of range in step 31, indicating that the symbol does not form part of a coded sequence, the procedure goes to step 41 where the status of the marker 'codeword_started' is verified. If this is true, this means that a codeword has been started, but the subsequent symbol does not form

20 part of the coded sequence. In step 42, therefore, the memory address is sent as the codeword. The memory address is then reset to 0 in step 43, the marker 'codeword_started' reset to false in step 44 and the procedure returns to step 31, where the fetched symbol is checked against the starting

symbols of coded sequences to verify if it forms part of a coded sequence. If in step 41, it is determined that no codeword has been started, i.e. the status of the 'codeword_started' marker is false, this means that the fetched symbol is not contained in any coded sequence. In step 45, the transmission mode is checked. If the last data sent was compressed, the symbol indicating uncompressed data is sent in step 46 followed by the read symbol in step 47. If the last transmission was not in compression mode, the symbol is sent in step 47. The procedure then returns to the start at step 30.

Decompression is the exact reverse of the compression procedure described above. Each codeword is converted into the corresponding symbol or symbol sequence. The symbols are then summed to retrieve the original uncompressed data.

Figs. 11a to 11e illustrate an example using the coding algorithm described above. The algorithm was first trained, i.e. the codebook generated, using an IEGM signal containing 10000 samples sampled at 512 Hz with 8-bit resolution. The training signal is illustrated in Fig. 11a. The algorithm was then tested on an IEGM signal containing 10000 samples sampled at 512 Hz with 8-bit resolution. This uncompressed test signal is depicted in Fig. 11b. Fig. 11c shows the signal after compression. This signal contains 2149 samples which gives a compression ratio of 4.6. Fig. 11d shows the signal of Fig. 11c after decompression. Finally, Fig. 11e shows the difference signal between the original signal depicted in Fig. 11b

and the decompressed signal of Fig. 11d. The signal is entirely free of distortion.

In the coding procedures described above, a codeword is allocated to every node and end branch of a probability tree. It will, however, be understood that codewords could be assigned only to the end branches of the tree.

Although modifications and changes may be suggested by those skilled in the art, it is the intention of the inventor to embody within the patent warranted hereon all changes and modifications as reasonably and properly come within the scope of his contribution to the art.

Compression and decompression coding scheme and apparatusField of invention

5 The invention relates generally to a coding scheme and apparatus for the compression and decompression of data. It is particularly directed to the compression of signals that exhibit so-called memory, where a portion of a signal depends on the value of a preceding portion. The invention has particular application to medical systems, such as implantable pacemaker
10 devices, which have limited memory but require the storage of large quantities of data.

Background art

15 Medical systems for the monitoring of physiological functions are becoming more complex as the need for diagnostic applications increases. In particular there is a need for intracardial detection systems and pacemaker systems capable of storing ever increasing numbers of signals, such as electrocardiogram signals (ECG and IEGM), pressure signals and bioimpedance signals or the like, of ever increasing length. However, the
20 available memory space for data storage is often restricted, particularly in implanted pacemaker systems. Perhaps more importantly in implanted systems, the amount of data that may be collected is also restricted by the transmission capacity of a telemetry link between an implanted device and its programmer or other external control device. For example, a defibrillator
25 today typically requires a transmission time of up to 40 minutes for downloading to its controller all the data that can be collected. If the required quantities of data are to be made available for processing, this data must be compressed.

30 ~~Data compression can generally be divided into two forms. A first form is~~

Data compression can generally be divided into two forms. A first form is based on viewing a signal as a mathematical function and observing and utilising characteristics of this function to compress data. The second form
5 utilises coding theory and is based on the statistics of multiple discrete signal levels, or symbols, occurring in a signal.

An algorithm of the second type is the Huffman code, which is discussed in US 5,448,642. According to the Huffman code, input data is parsed into fixed
10 length sequences, which are then coded with variable length codewords depending on their probability. A further conventional algorithm working according to this latter principle is the Tunstall code. This code maps variable-length symbol segments of an input signal into fixed-length codewords. Since the codeword length is fixed, the number of codewords n is known in advance.
15 The object is to assign codewords to symbol segments that occur with approximately equal probability. The procedure begins with a set of symbol segments consisting of each of the individual symbols occurring in the input signal, say m symbols in all. The most probable symbol is then removed from the set and replaced by m new segments, each of which is the removed symbol
20 suffixed by one of the m input symbols. This procedure continues until the number of symbol segments, m , in the set is equal to the number of codewords, n . The codewords are then assigned to the symbol segments.

An example of Tunstall encoding is illustrated in Figs. 1 and 2. Fig. 1 shows a
25 probability tree comprising nodes, and branches emanating from each node. In the illustrated example it is assumed that 8 codewords are available for compressing data, 1 to 8. The tree in Fig. 1 assumes that a source signal comprises two signals, '0' and '1'. Each symbol is represented by a first branch. The probability a symbol occurring is given at the node terminating

the associated branch. Hence a '0' occurs with a probability of 0.6 and a '1' occurs with a probability of 0.4. The branch with the highest probability is expanded further by adding a second series of branches for each possible symbol. The '0' branch is thus bifurcated into a second '0' branch with a total probability of 0.36 and a '1' branch with probability 0.24. The first branch for symbol '1' now has the highest probability and is expanded in turn, resulting in two further branches with probability of 0.24 and 0.16. This process continues until the number of end branches equals the number of codewords, in the present example 8, with the probability of each branch occurring being relatively close. Each end branch represents a sequence of symbols. These sequences are assigned a codeword as shown in Fig. 2.

A problem associated with Tunstall encoding is that signals containing a large number of different symbols, for instance a large number of discrete signal levels, require a very large number of codewords. For example, in a pacemaker system, a typical electrogram or IEGM signal is represented by 8 bits sampled at 512 Hz. The number of symbols in this signal is thus 256. A Tunstall codebook for this signal would have to contain at least 256 codewords just to cover the different individual symbols. In order to obtain compression of the signal, the branches must be expanded further which adds a further 256 codewords per branch. The Tunstall code is a general-purpose data compression algorithm and is not adapted to special classes of data. In particular it is not effective when applied to signals which exhibit memory. Typically, many of the signals monitored by medical systems, and cardiac pacers in particular, exhibit some memory, one example being the IEGM signal. The use of Tunstall encoding for processing sampled data in medical systems, and specifically implanted systems, is thus limited.

In the light of this prior art, it is an object of the present invention to provide a

coding method and apparatus that allows a signal exhibiting memory to be compressed and decompressed efficiently and without distortion.

SUMMARY OF INVENTION

5 Viewed from one aspect, the present invention proposes a data coding method for converting a signal containing a plurality of symbols into codewords, including the steps of : establishing a set of fixed-length codewords, observing a data signal and determining the most frequently occurring symbols and/or sequences of symbols containing at least two symbols, allocating one
10 codeword to each of the most frequently occurring of said symbols and/or symbol sequences and reserving at least one codeword to serve as indicator for uncoded symbols.

15 According to a further aspect, the invention provides a data compression method for compressing a data signal containing a plurality of symbols, including: converting the most frequently occurring symbols and/or symbol sequences into one of a set of fixed-length codewords, supplementing the remaining symbols with at least one codeword indicative of no compression.

20 The invention further proposes an arrangement for compressing and decompressing a data signal containing a plurality of symbols, including: means for storing fixed-length codewords corresponding to symbols and/or symbol sequences with at least one codeword reserved for indicating no compression, and means for determining if a symbol in said data signal
25 corresponds to at least one codeword in the storage means and, when a symbol corresponds to only one codeword, for transmitting said codeword, wherein the determining means are further adapted to transmit a symbol supplemented by said at least one reserved codeword if it corresponds to no codeword in the storage means.

5 According to a fourth aspect, the invention proposes a computer program product for converting a signal containing a plurality of symbols into a compressed signal, including computer readable program code means for establishing a set of fixed-length codewords, determining the most frequently occurring symbols and/or sequences of symbols containing at least two symbols in a data signal, allocating one codeword to each of the most frequently occurring of said symbols and/or symbol sequences and reserving at least one codeword to serve as indicator for uncoded symbols.

10

By providing codewords to only the most frequently occurring symbols and symbols sequences, the number of codewords required is greatly reduced. At the same time however the efficiency of the compression is increased, as these

codewords are allocated to the varying length symbol sequences that appear with the highest frequency in the signal. This compression technique furthermore fully exploits any memory in a signal. A further difference over prior art coding schemes and the Tunstall code in particular is that codewords may be allocated to every node and end branch in a coding tree and not just the end branches. This is also particularly useful when compressing signals that exhibit memory such as signals monitoring physiological quantities such as the heartbeat, respiration rate and the like.

BRIEF DESCRIPTION OF THE DRAWINGS

Further objects and advantages of the present invention will become apparent from the following description of the preferred embodiments that are given by way of example with reference to the accompanying drawings, in which:

Fig. 1 depicts a coding tree illustrating the Tunstall encoding algorithm;

Fig. 2 depicts a coding table corresponding to the coding tree of Fig. 1

Fig. 3 depicts a schematic block diagram of an encoder/decoder (compressor/decompressor) according to the present invention.

Figs. 4a depicts a series of histograms illustrating the generation of a codebook according to the present invention;

Fig. 5 depicts a coding tree corresponding to the histograms of Fig. 3;

Fig. 6 depicts a codebook corresponding to the tree of Fig. 4;

Fig. 7 is a schematic illustration of a pre-processing function in

accordance with the present invention.

Fig. 8 is a schematic diagram showing an arrangement for compressing and decompressing data according to the present invention;

Fig. 9 is a schematic illustration of a codebook memory illustrating the mapping between symbols and memory locations;

Fig. 10 is a flow chart illustrating a method for compressing data utilising the arrangement of Fig. 8; and

Figs 11a to 11e show a sequence of graphs illustrating the compression and subsequent decompression of an IEGM signal.

DETAILED DESCRIPTION OF THE DRAWINGS

The compressing and decompression scheme according to the present invention is based on the representation of a data signal containing a plurality of symbols into coded form utilising codewords of fixed length. The symbols contained in an input signal may be digital representations of characters, such the ASCII format. Typically, however, the symbols will be binary representations of discrete signal levels in a sampled analogue signal, such as an ECG, IEGM or other signals for monitoring physiological activity.

Fig. 3 schematically illustrates the function of an encoder/decoder for compressing and decompressing a data signal. The data signal contains a plurality of symbols which are read by a symbol reader 1 and relayed to an encoder/decoder 2. The encoder/decoder 2 has access to a storage medium 3 containing a codebook of the fixed-length codewords. The input symbols or symbol sequences are mapped to codewords in the codebook 3 and replaced by the corresponding codeword. The reduction of symbols and symbol

sequences of variable length into codewords results in the compression of the data. Decompression is accomplished by inverting the operation. Codewords are passed through the encoder/decoder 2, which with the aid of the mapping to the codebook 3 reconstitutes the original input sequence.

5

The generation of a codebook according to the present invention commences with determining the number of codewords to be used. This is selected as a function of the total number of symbols contained in an input signal and the degree of compression required.

10

The codebook is generated by observing the input signal during a test phase and determining the probabilities of the symbols and sequences of symbols occurring. This is accomplished by observing the input signal during a set time period, for example 20 s, and noting the symbol that occurs with the highest frequency during this period. The signal is then observed again for the same time period and the symbol that follows the noted symbol most frequently is noted. This process continues until the number of most frequently occurring symbols and/or symbol sequences noted is equal to the number of codewords in the codebook. It will be understood, that only those symbols or symbol sequences that occur most frequently will be coded. This allows the number of codewords to be kept to a reasonable number.

15

20

A simplified example of this process is illustrated in Figs. 4, 5 and 6 with reference to the generation of a codebook containing 8 symbols. The input signal consists of 5 symbols 1,2,3,4,5.

25

Turning now to Fig. 4a, a first histogram is produced after observing the input signal for a set period. It is evident that the most frequently occurring symbol is 3 with an occurrence of 50. Fig. 4b shows a histogram produced after the same period of time and gives the frequency of the different symbols that

30

follow 3. Thus the sequence 32 has occurred ten times, the sequence 33 fifteen times, and the sequence 34 twelve times. Fig. 4c shows a histogram of the symbols following 2 and Fig. 4d shows a histogram of the symbols following the symbol sequence 3, 3. Finally Fig. 4e shows the histogram of the symbols following the symbol 4. The codebook is established by determining which of the 8 symbols and/or symbol sequences occurred most frequently. This is illustrated schematically in a code tree in Fig. 5. In this code tree, a codeword has been assigned to every node and end branch. Hence the symbol 3 has been allocated the codeword '2' but the symbol sequence 33 has also been allocated a codeword, '5' and the symbol sequence 332 has been allocated the codeword '8'.. This is summarised in tabular form in Fig. 6.

It is apparent from the above example that codewords are not assigned to every symbol of an input signal. Thus when these unassigned symbols are read by an encoder, the symbol is transmitted uncompressed. To distinguish the uncompressed data from the compressed data, it is preferably to utilise some form of distinguishing symbol or codeword. Thus at least one codeword will be reserved for transmitting uncoded input symbols.

Depending on the signal to be compressed, it may be desirable to pre-process the signal in order to increase the frequency of a very few symbols and sequences. This considerably increases the efficiency of the compression. For example, pre-processing can be very effective when compressing sampled IEGM, ECG or other physiological signals that monitor some form of periodic activity.

Fig. 7 schematically illustrates the preferred function of such a pre-processor. In Fig. 7 an input data stream denoted by 10 comprises the symbols IEGM_n-1, IEGM_n, IEGM_n+1 and IEGM_n+2. The function generates the difference symbol value between a symbol and a preceding symbol. The output data

stream 12 thus comprises the symbols (IEGMn - IEGMn-1) and (IEGMn-1 - IEGMn-2), etc.

In a preferred embodiment of the present invention, the compression coding scheme according to the invention is used to compress an IEGM signal comprising 255 symbols ranging from 0 to 254. After pre-processing in accordance with the function of Fig. 7, the signal contains 509 possible symbols, ranging from -254 to +254. However, while the number of possible symbols has almost doubled, the form of the original IEGM signal is such that the processed signal contains mainly symbols close to 0 such as 1, -1, 2, -2 etc. Thus the concentration of a very few symbols and symbol sequences is increased by the difference function.

A codebook generated for a pre-processed IEGM signal of the type described above can typically be efficiently compressed utilising a codebook containing 254 codewords, for example 254 8-bit words ranging from 0 to 253. The 254 most frequently occurring symbols and/or symbol sequences are then converted into codewords. The 8-bit words with values 254 and 255 are then reserved. These are utilised to signal whether data is compressed or uncompressed.. Preferably, one reserved codeword, for example 255, is sent to indicate that uncompressed data follows. The other codeword 254 can then be utilised to signal that compressed data is following. To avoid having to generate different symbols for negative and positive symbols, an uncompressed negative symbol is preferably indicated by preceding it with both symbols sent contiguously, for example, 255 followed by 254 followed by the uncompressed equivalent positive symbol.

The codebook is preferably generated for a class of signals and retained for different signals of the same class so that it does not need to be re-established prior to compressing data. For example, a cardiac pacer containing data

compression and decompression circuitry for processing some physiological measurement such as an IEGM signal, ECG signal or bioimpedance signal could be subjected to a training phase for each patient to establish a codebook that is specific to each patient. The training phase could also be performed using a test sequence that is representative of the class of signal, so that one codebook is used for several patients. A further option would be to utilise an adaptive procedure, whereby the statistics of the signal are observed and a codebook newly generated prior to each compression. This codebook would then be optimised for each signal generated in a pacemaker. However, it will be understood that the codebook must be retained long enough to enable the compressed data to be decompressed.

In the coding procedures described above, a codeword is allocated to every node and end branch of a probability tree. It will, however, be understood that codewords could be assigned only to the end branches of the tree. The efficiency of such a code would depend on the characteristics of the starting signal.

An arrangement for encoding and decoding a signal is illustrated in Fig. 8. The arrangement includes an input stage 20 for reading a symbol. A processor 21, that preferably takes the form of a single chip microprocessor with associated memory, is coupled to the input stage. A codebook memory 22 having 256 8-bit memory locations with addresses from 0 to 255 is connected to the processor 21. An output stage 23 for the codewords is coupled to the memory 22 and processor 21. This output stage 23 is finally coupled to a storage memory 24 for storing the compressed data and to a telemetry transmission unit 26, which forwards the data to a remote external programmer or controller. The input stage 20 is also coupled to the storage memory 24 and telemetry transmission unit 26 for transmitting uncompressed data. The encoding and decoding arrangement is preferably preceded by a pre-

processing stage as shown in Fig. 7.

The function of this arrangement is basically as follows. A first symbol is read by the input stage 20. The processor 21 checks whether this symbol corresponds to more than one symbol sequence in the codeword memory 22 and if so, the next symbol is read. This process is repeated until the sequence of symbols read corresponds to only one codeword. This codeword is then emitted in place of the symbol sequence and is either stored in the memory 24 or sent directly to an external device via the telemetry transmission unit 26. If a symbol read corresponds to no coded sequences it is transmitted unchanged to the storage memory 24 or transmission unit 26 but preceded by the codeword 254 indicating uncompressed data.

The addresses 0 to 255 of the codebook memory 22 are the codewords. The processor furthermore performs a mapping between the incoming symbols that form at least part of a coded sequence and each memory cell 221. This is illustrated in more detail in Fig. 9. The first symbol in a sequence will cause the processor to access the first memory cell, in ascending order of address, to which the symbol is mapped. These first memory cells 221 correspond to the first branches in a probability tree. Thus if the first symbol in a sequence is '1', the processor 21 will access the fourth memory location (address 3), since this is the first location 221 to which a '1' is mapped. Each memory cell 221 further contains information indicating whether the mapped symbol corresponds to more than one coded symbol sequence. This information is shown on the right hand side of each memory cell 221. The information is in the form of a code represented by the numbers 0 to 7, which indicate both the number of possible further branches and the following symbols corresponding to the further branches. A conversion table 25 shows the significance of each number. Hence it can be seen that '0' indicates an end branch, '2' represents two possible further branches with the following symbols '1' and '2'. Any

symbol mapped to a memory cell 221 containing a '0' will result in the processor transmitting the address of the memory cell as a codeword. Any other number contained in the memory cell 221 indicates that at least one further branch is possible. The processor 21 then fetches the next symbol from the input stage 20 and determines whether this symbol forms part of the possible codewords. If it does, the processor calculates the address of the next memory cell 221 and the process is repeated.

The next address is calculated by summing the number of possible branches contained in all addresses starting from the first address up to the present address, and then adding the position of the fetched symbol in the list of possible symbols given in the conversion table 25. The sum of possible branches is equal to the sum of the first mapped memory locations representing the initial branches of the probability tree and the possible branches stored in all previous memory locations. Thus an input sequence consisting of 1, 2 would result in a first mapping being made to memory location 3. This contains the code 5, which indicates that four further branches are possible. The next symbol is fetched. It is verified that '2' corresponds to one of the possible branches. The symbol '2' has the fourth position in the list of branches (-1, 0, 1, 2) given in the conversion table 25. Thus the sequence 1, 2 is a valid coded sequence. The next address is equal to the sum of the initial basic branches (i.e. the addresses to which a first symbol can be mapped), which in the present example is 5 (addresses 0 to 4). To this is added the sum of the branches contained in memory locations up to address 3, and the position of the subsequent symbol in the list indicated in memory location 3. This gives a total of $5 + (3 + 1 + 2) + 4 = 15$. The next address is thus the 15th location or address 14, since the addresses start with 0. This is verified by the mapping of the symbol 2 indicated in the left-hand column of the cell with address 14.

The conversion table 25 may alternatively be more complex in structure and provide absolute memory locations corresponding to each possible branch. In this way the next address would not need to be calculated, but more storage capacity would be needed.

The conversion table is preferably stored in the processor 21. However, it may be possible to store some of the information about the branches in the codeword memory, depending on how much capacity is available for each address.

The arrangement illustrated in Fig. 8 is a schematic representation of possible encoding and decoding hardware. It will however, be understood that the functions of the various elements shown in Fig. 8 may be performed entirely in a digital processor system operating under the control of a program. The codebook memory could then be implemented virtually from part of the memory space incorporated in the processor system.

Fig. 10 is a flow chart illustrating the procedure for compressing data using the arrangement shown in Figs. 8 and 9. The procedure starts at step 30 with the reading of a symbol. In step 31 it is determined whether the symbol is inside the designated range, i.e. whether the symbol forms part of a coded sequence and can be mapped to a memory location. If this is the case, the process moves to step 32 and the new memory address is calculated. In the following step 33 a marker, 'codeword_started', indicating that the coding of a sequence has started, is set. In step 34 it is verified whether the data last sent was uncompressed. If this is true, in step 35 the symbol indicating compression is sent. If the last data sent was in compressed mode, the process moves directly from step 34 to step 36, where the contents of the memory location are read and it is determined whether an end branch has been reached. If the end branch has not been reached the procedure returns to step 30 and the next

symbol is fetched. If the end branch is reached, the procedure passes to step 38, where the memory address is sent as the codeword. In step 39, the memory address is reset to 0 and in step 40, the marker 'codeword_started' is reset to false, because the coding of a symbol or symbol sequence is terminated. The process then returns to step 30, and the next symbol fetched. If a symbol is discovered to be out of range in step 31, indicating that the symbol does not form part of a coded sequence, the procedure goes to step 41 where the status of the marker 'codeword_started' is verified. If this is true, this means that a codeword has been started, but the subsequent symbol does not form part of the coded sequence. In step 42, therefore, the memory address is sent as the codeword. The memory address is then reset to 0 in step 43, the marker 'codeword_started' reset to false in step 44 and the procedure returns to step 31, where the fetched symbol is checked against the starting symbols of coded sequences to verify if it forms part of a coded sequence. If in step 41, it is determined that no codeword has been started, i.e. the status of the 'codeword_started' marker is false, this means that the fetched symbol is not contained in any coded sequence. In step 45, the transmission mode is checked. If the last data sent was compressed, the symbol indicating uncompressed data is sent in step 46 followed by the read symbol in step 47. If the last transmission was not in compression mode, the symbol is sent in step 47. The procedure then returns to the start at step 30.

Decompression is the exact reverse of the compression procedure described above. Each codeword is converted into the corresponding symbol or symbol sequence. The symbols are then summed to retrieve the original uncompressed data.

Figs. 11a to 11e illustrate an example using the coding algorithm described above. The algorithm was first trained, i.e. the codebook generated, using an IEGM signal containing 10000 samples sampled at 512 Hz with 8-bit

resolution. The training signal is illustrated in Fig. 11a. The algorithm was then tested on an IEGM signal containing 10000 samples sampled at 512 Hz with 8-bit resolution. This uncompressed test signal is depicted in Fig. 11b. Fig. 11c shows the signal after compression. This signal contains 2149 samples which gives a compression ratio of 4.6. Fig. 11d shows the signal of Fig. 11c after decompression. Finally, Fig. 11e shows the difference signal between the original signal depicted in Fig. 11b and the decompressed signal of Fig. 11d. The signal is entirely free of distortion.

In the coding procedures described above, a codeword is allocated to every node and end branch of a probability tree. It will, however, be understood that codewords could be assigned only to the end branches of the tree.

P000696 SE/HG

Claims:

- 5 1. A data coding method for converting an input signal containing a plurality of symbols into a compressed signal, including establishing a set of fixed-length codewords, the method being characterised by the steps of:
observing a data signal and determining the most frequently occurring symbols and/or sequences of symbols containing at least two symbols,
10 allocating one codeword to each of the most frequently occurring of said symbols and/or symbol sequences, and
reserving at least one codeword in the set to serve as indicator for uncoded symbols.
- 15 2. A method as claimed in claim 1, characterised in that each determining step is performed by observing the data signal during a predetermined time period.
- 20 3. A method as claimed in claim 1 or 2, characterised by allocating codewords to symbols and symbol sequences that are incorporated in other symbol sequences having an allocated codeword.
- 25 4. A method as claimed in any one of claims 1 to 3, characterised by supplementing uncoded negative symbols with at least one codeword indicative of a negative value.
5. A data compression method for compressing a data signal containing a plurality of symbols, characterised by including:
converting the most frequently occurring symbols and/or symbol

sequences into one of a set of fixed-length codewords (38, 42),
supplementing the remaining symbols with at least one codeword
indicative of no compression (46, 47).

- 5 6. A method as claimed in claim 5, wherein the number and length of the
codewords are predetermined.
- 10 7. A method as claimed in claim 5 or 6, including pre-processing an input
signal containing a plurality of symbols to generate said data signal by
generating a symbol representing the difference between contiguous
symbols.
- 15 8. A method as claimed in any one of claims 5 to 7 further characterised
by
reading a symbol (30),
determining if said symbol corresponds to at least one codeword (31,
41), and
outputting a codeword if said symbol corresponds to only said one
codeword .
- 20 9. A method as claimed in claim 8, characterised in that when a symbol
corresponds to more than one codeword, reading a subsequent symbol,
determining if said symbol corresponds to at least one codeword, and
outputting a codeword if said symbol corresponds to only said one
codeword.
- 25 10. A method as claimed in claim 8 or 9, characterised by
when a symbol corresponds to no codeword, outputting said symbol.

11. An arrangement for compressing and decompressing a data signal containing a plurality of symbols, characterised by including:
means (22) for storing fixed-length codewords corresponding to symbols and/or symbol sequences with at least one codeword reserved for indicating no compression, and
5 means (20, 21, 23, 25) for determining if a symbol in said data signal corresponds to at least one codeword in said storage means and, when a symbol corresponds to only one codeword, for transmitting said codeword, wherein the determining means are further adapted to
10 transmit a symbol supplemented by said at least one reserved codeword if said symbol corresponds to no codeword in said storage means.
12. An arrangement as claimed in claim 11, characterised in that said storage means (22) include a plurality of storage locations (221)
15 designating codewords, wherein each storage location (221) contains an indication of the number of possible coded symbol sequences, and is mapped to a symbol of said data signal.
13. An arrangement as claimed in claim 11 or 12, characterised in that
20 means generating a difference symbol between contiguous symbols in said data signal are connected upstream of said determining means (20, 21, 23, 25).
14. An arrangement as claimed in any one of claims 11 to 13, characterised
25 in that the addresses of the storage locations (221) are codewords.
15. A computer program product for converting a signal containing a plurality of symbols into a compressed signal, characterised by including computer readable program code means for establishing a set

of fixed-length codewords, determining the most frequently occurring symbols and/or sequences of symbols containing at least two symbols in a data signal,
allocating one codeword to each of the most frequently occurring of
5 said symbols and/or symbol sequences, and
reserving at least one codeword to serve as indicator for uncoded symbols.

16. A program product as claimed in claim 15, further characterised by
10 computer readable program code means for compressing said data signal by converting the most frequently occurring symbols and/or symbol sequences into said fixed-length codewords by:
reading a symbol,
determining if said symbol corresponds to at least one codeword, and
15 outputting a codeword if said symbol corresponds to only said one codeword, and by outputting a symbol when said symbol corresponds to no codeword,

20

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
15 March 2001 (15.03.2001)

PCT

(10) International Publication Number
WO 01/18973 A1

(51) International Patent Classification⁷: **H03M 7/30,**
A61B 5/0402

[SE/SE]; Krukmakargatan 63 kv, S-117 41 Stockholm (SE).

(21) International Application Number: PCT/SE00/01744

(74) Common Representative: **ST. JUDE MEDICAL AB;**
Patent Department, S-175 84 Järfälla (SE).

(22) International Filing Date:
7 September 2000 (07.09.2000)

(81) Designated State (*national*): US.

(25) Filing Language: English

(26) Publication Language: English

(84) Designated States (*regional*): European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).

(30) Priority Data:
9903191-6 8 September 1999 (08.09.1999) SE

Published:
— With international search report.

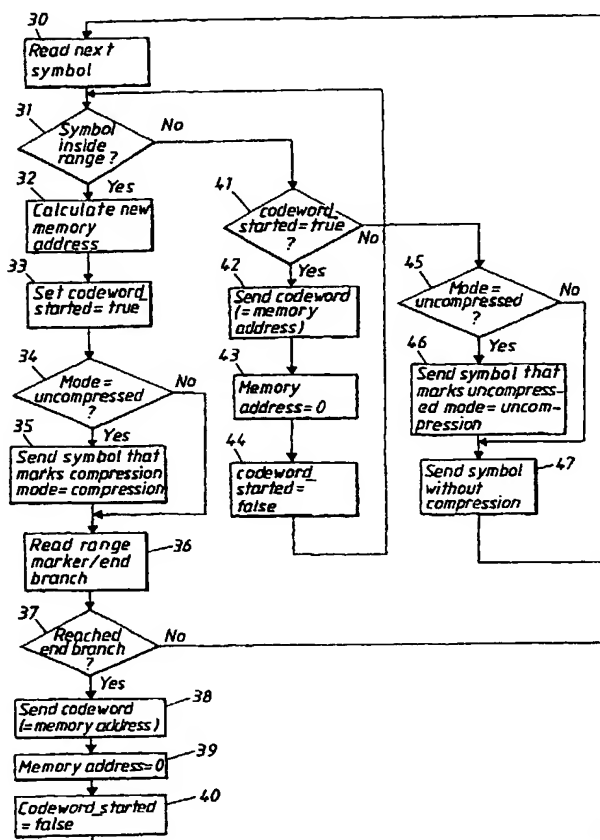
(71) Applicant (for all designated States except US): **ST. JUDE MEDICAL AB** [SE/SE]; S-175 84 Järfälla (SE).

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(72) Inventor; and

(75) Inventor/Applicant (for US only): **LIDMAN, Johan**

(54) Title: COMPRESSION AND DECOMPRESSION CODING SCHEME AND APPARATUS



(57) Abstract: The invention relates to a compression and decompression coding scheme and arrangement. The scheme converts a data signal containing a plurality of symbols into a series of codewords. A set of codewords is established and the data signal is subsequently observed to determining the most frequently occurring symbols and/or sequences of symbols containing at least two symbols. A codeword is then allocated to each of the most frequently occurring of said symbols and/or symbol sequences. At least one codeword is reserved for indicating uncompressed data. When compressing a signal, the incoming symbols are first checked to see if they correspond to at least one codeword. If a symbol corresponds to more than one codeword, further symbols are read until a symbol is found which corresponds to one codeword only. The codeword is then transmitted. Any symbols that correspond to no codewords are supplemented with a codeword indicative of no compression and then transmitted.

WO 01/18973 A1

Fig. 1

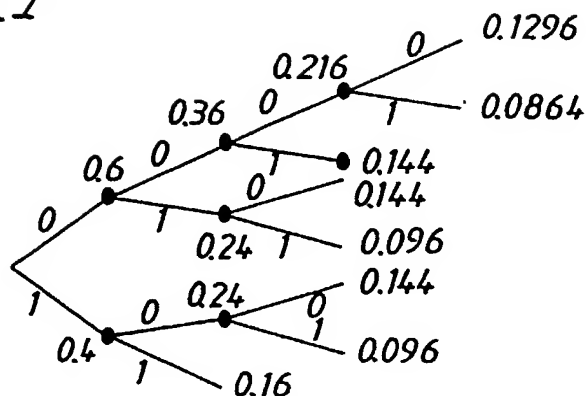
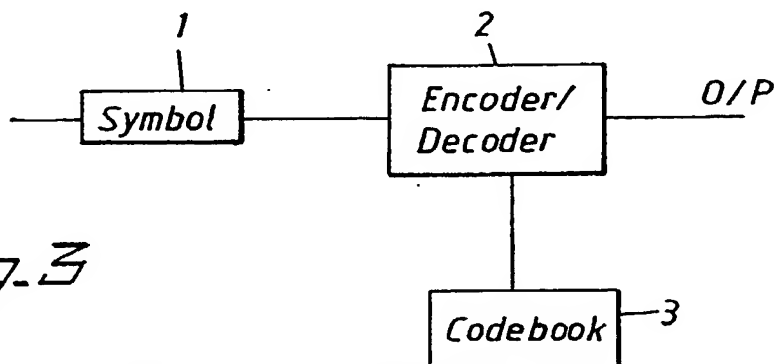


Fig. 2

Symbol sequences	Codeword	Codeword binary form
0000	1	001
0001	2	010
001	3	011
010	4	100
011	5	101
100	6	110
101	7	111
11	8	000

Fig. 3



2 / 6

Fig. 4a

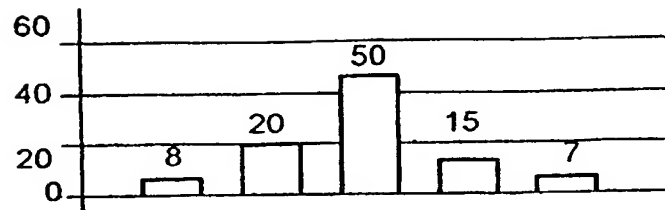


Fig. 4b

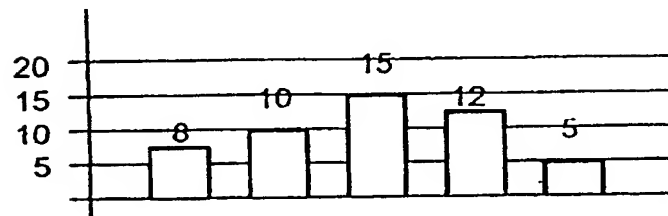


Fig. 4c

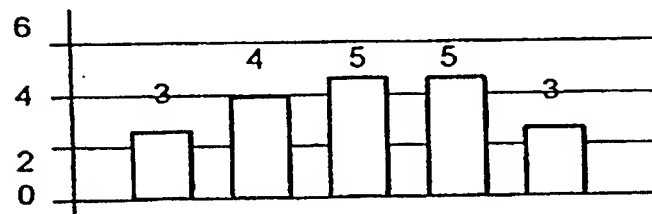


Fig. 4d

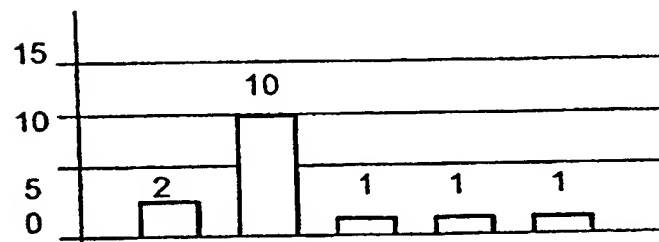


Fig. 4e

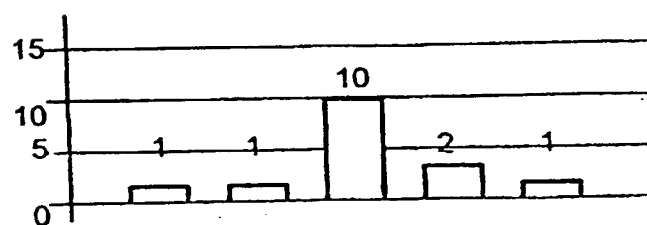


Fig. 5

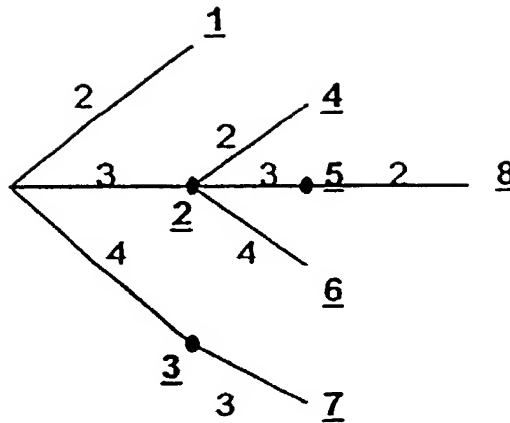
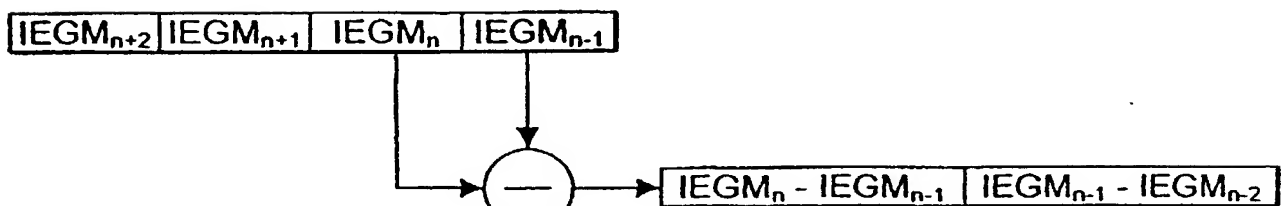


Fig. 6

Symbol sequence	Codeword
2	1
3	2
4	3
32	4
33	5
34	6
43	7
332	8

Fig. 7



4 / 6

Fig. 8

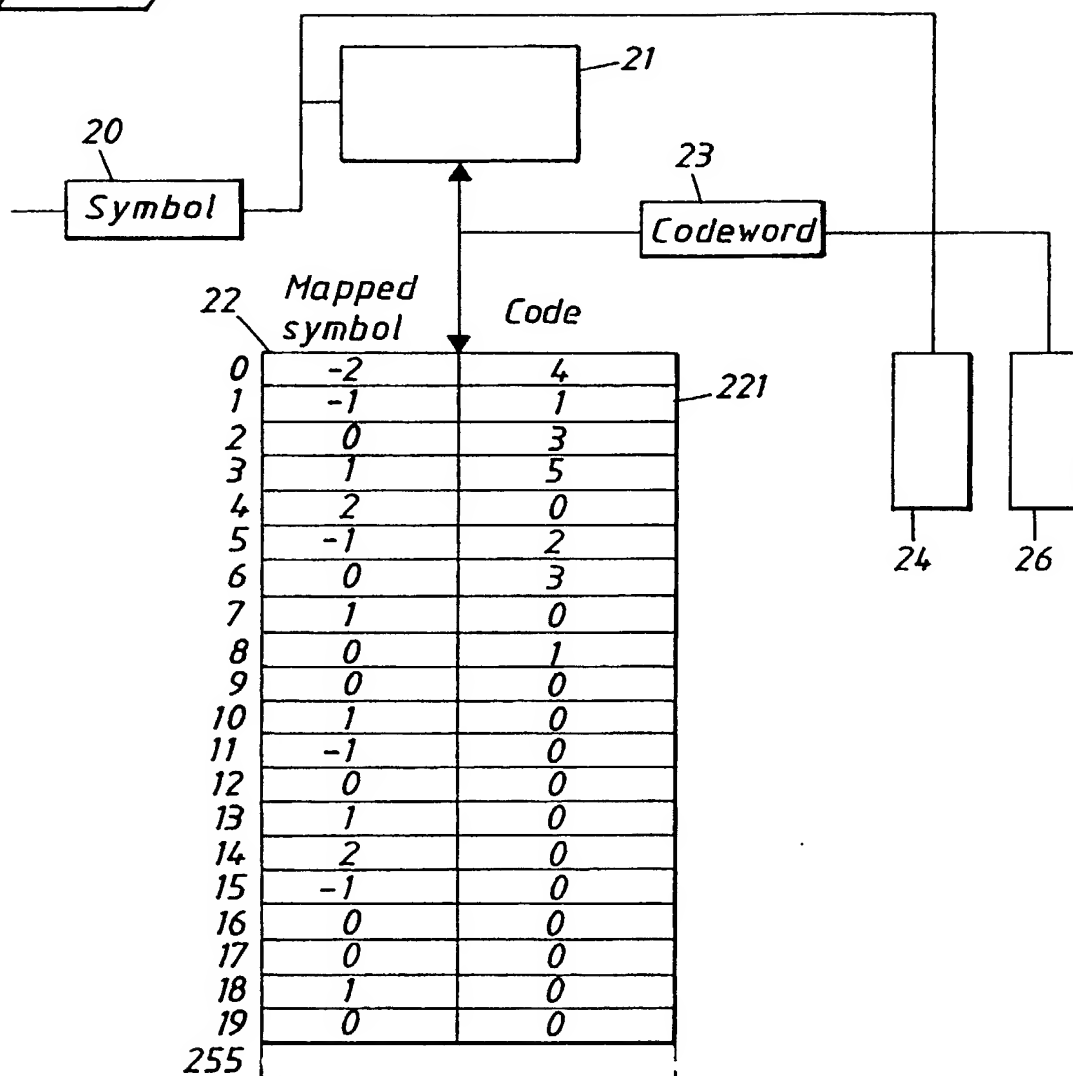


Fig. 9

25

Code No.	Possible branches	No. of branches
0	End	0
1	0	1
2	-1, 0	2
3	0, 1	2
4	-1, 0, 1	3
5	-1, 0, 1, 2	4
6	-2, -1, 0, 1	4
7	-2, -1, 0, 1, 2	5

5 / 6

Fig. 10

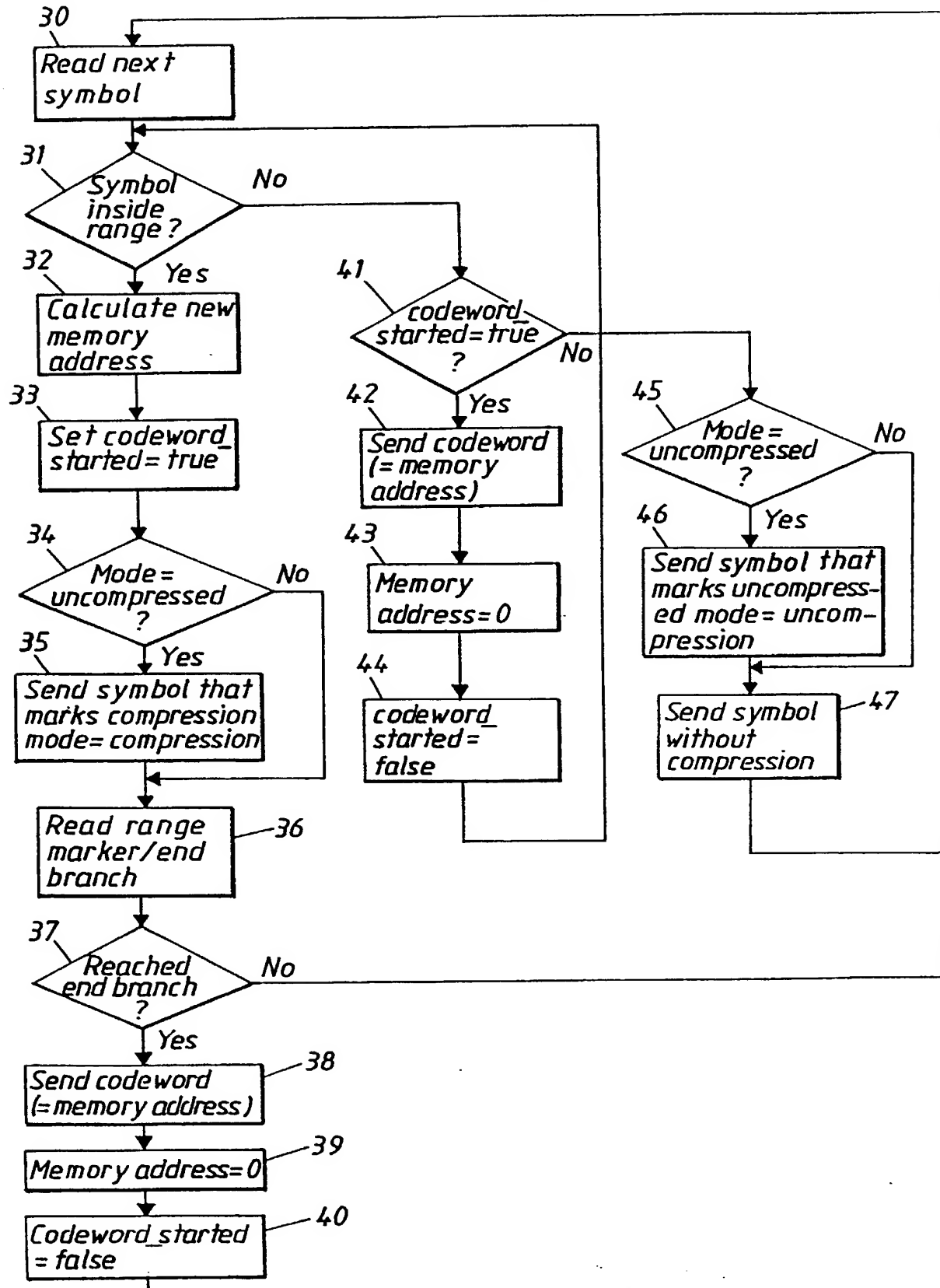


Fig. 11a

616

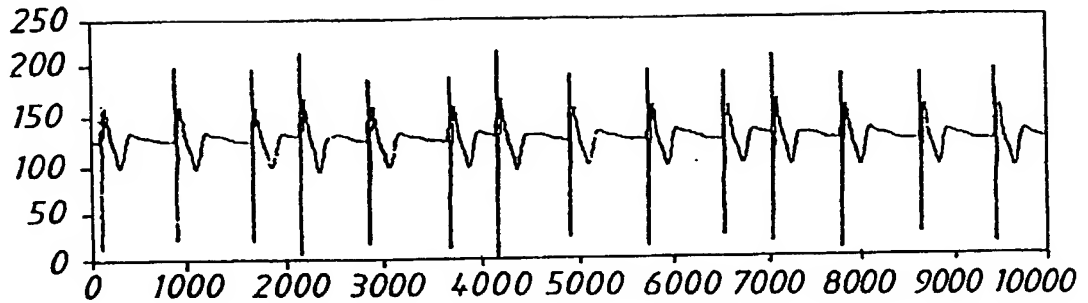


Fig. 11b

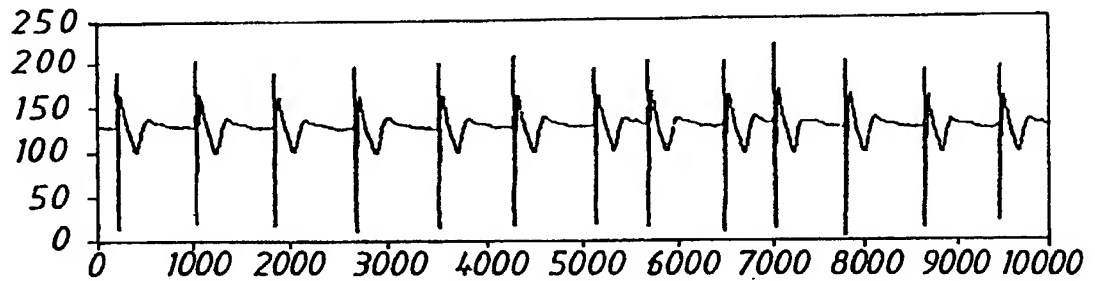


Fig. 11c

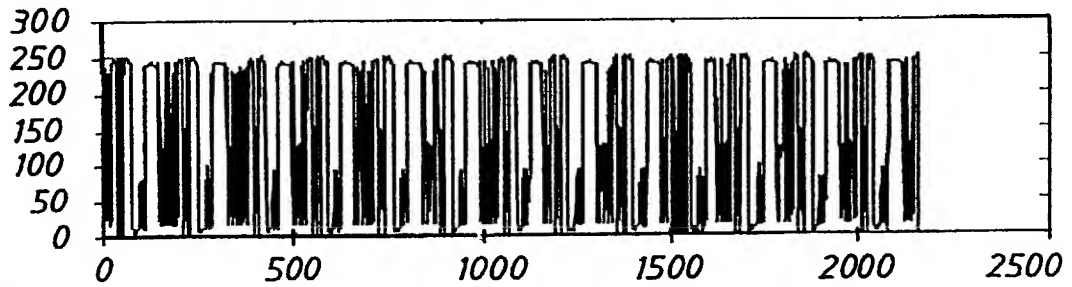


Fig. 11d

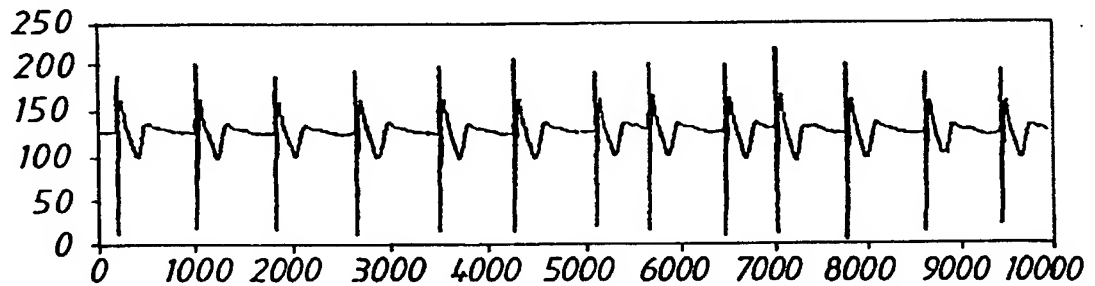
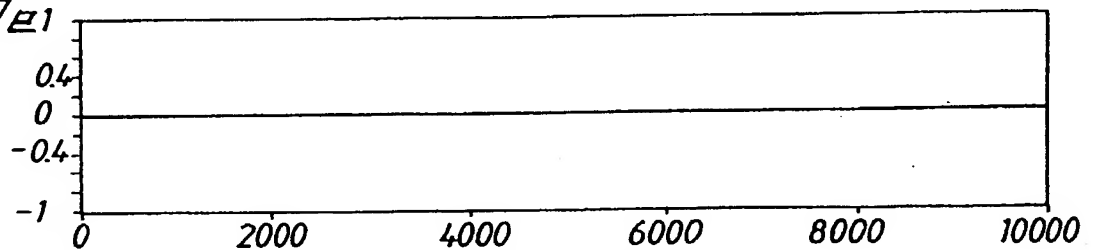


Fig. 11e1





10070701, 070702

#4.

COMBINED DECLARATION FOR PATENT APPLICATION AND POWER OF ATTORNEY (Includes Reference to PCT International Applications)	ATTORNEY'S DOCKET NUMBER P02,0086		
<p style="text-align: center;">As a below named inventor, I hereby declare that:</p> <p>My residence, post office address and citizenship are as stated below next to my name, I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled:</p> <p style="text-align: center;">"COMPRESSION AND DECOMPRESSION CODING SCHEME AND APPARATUS"</p> <p>the specification of which (check only one item below):</p> <div style="margin-left: 20px;"><p><input type="checkbox"/> is attached hereto.</p><p><input checked="" type="checkbox"/> was filed as United States application Serial No. <u>10/070,701</u> on <u>March 7, 2002</u> and was amended on <u>March 7, 2002</u> (if applicable).</p><p><input type="checkbox"/> was filed as PCT international application Number _____ on _____ and was amended under PCT Article 19 on _____ (if applicable).</p></div> <p>I hereby state that I have reviewed and understand the contest of the above-identified specification, including the claims, as amended by any amendment referred to above.</p> <p>I acknowledge the duty to disclose information which is material to the examination of this application in accordance with Title 37, Code of Federal Regulations, §1.56(a).</p> <p>I hereby claim foreign priority benefits under Title 35, United States Code, §119 of any foreign application(s) for patent or inventor's certificate or of any PCT international application(s) designating at least one country other than the United States of America listed below and have also identified below any foreign application(s) for patent or inventor's certificate or any PCT international application(s) designating at least one country other than the United States of America filed by me on the same subject matter having a filing date before that of the application(s) of which priority is claimed:</p>			
PRIOR FOREIGN/PCT APPLICATION(S) AND ANY PRIORITY CLAIMS UNDER 35 U.S.C. 119:			
COUNTRY (if PCT indicate "PCT")	APPLICATION NUMBER	DATE OF FILING (day, month, year)	PRIORITY CLAIMED UNDER 35 USC 119
Sweden	9903191-6	8 September 1999	<input checked="" type="checkbox"/> YES <input type="checkbox"/> NO
			<input type="checkbox"/> YES <input type="checkbox"/> NO
			<input type="checkbox"/> YES <input type="checkbox"/> NO
			<input type="checkbox"/> YES <input type="checkbox"/> NO
			<input type="checkbox"/> YES <input type="checkbox"/> NO

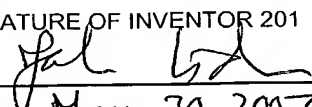
Combined Declaration For Patent Application and Power of Attorney (Continued) (Includes Reference to PCT International Applications)				ATTORNEY'S DOCKET NO. P02,0086	
I hereby claim the benefit under Title 35, United States Code, §120 of any United States application(s) or PCT international application(s) designating the United States of America that is/are listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in that/those prior application(s) in the manner provided by the first paragraph of Title 35, United States Code, §112, I acknowledge the duty to disclose material information as defined in Title 37, Code of Federal Regulations, §1.56(a) which occurred between the filing date of the prior application(s) and the national or PCT international filing date of this application:					
PRIOR U.S. APPLICATIONS OR PCT INTERNATIONAL APPLICATIONS DESIGNATING THE U.S. FOR BENEFIT UNDER 35 U.S.C. 120:					
U.S. APPLICATIONS				STATUS (Check one)	
U.S. APPLICATION NUMBER	U.S. FILING DATE		PATENTED	PENDING	ABANDONED
PCT APPLICATIONS DESIGNATING THE U.S.					
PCT APPLICATION NO	PCT FILING DATE	U.S. SERIAL NUMBERS ASSIGNED (if any)			

POWER OF ATTORNEY: As a named inventor, I hereby appoint the following attorney(s) and/or agent(s) to prosecute this application and transact all business in the Patent and Trademark Office connected herewith.

And I hereby appoint all Attorneys identified by the United States Patent & Trademark Office Customer Number 26574, who are all members of the firm of Schiff, Hardin & Waite.

Send Correspondence to: SCHIFF, HARDIN & WAITE Patent Department 6600 Floor Sears Tower, Chicago, Illinois 60606 Customer Number 26574				Direct Telephone Calls to: 312/258-5790	
1 - 00 2 0 1	FULL NAME OF INVENTOR	FAMILY NAME <u>LIDMAN</u>	FIRST GIVEN NAME <u>JOHAN</u>	SECOND GIVEN NAME	
	RESIDENCE & CITIZENSHIP	CITY <u>STOCKHOLM</u> <u>SEX</u>	STATE OR FOREIGN COUNTRY SWEDEN	COUNTRY OF CITIZENSHIP SWEDEN	
	POST OFFICE ADDRESS	POST OFFICE ADDRESS KRUUKMAKARGATAN 63 kv,	CITY S117-41 STOCKHOLM	STATE & ZIP CODE/COUNTRY S117-41 STOCKHOLM, SWEDEN	

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under section 1001 of Title 18 of the United States Code, and that such willful false statements may jeopardize the validity of the application or any patent issuing thereon.

SIGNATURE OF INVENTOR 201 	SIGNATURE OF INVENTOR 202	SIGNATURE OF INVENTOR 203
DATE <u>May 30, 2007</u>	DATE	DATE